

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
KOMPIUTERIJOS KATEDRA

Baigiamasis magistro darbas

WEB draudimo sistemų projektavimas ir kūrimas.

Atliko: 2M kurso studentai

Jelena Ptak

Aleksandras Kulačėnkovas

Darbo vadovas:

asist. Linas Būtėnas

Vilnius
2007

Turinys

Paveiksliukų sąrašas	3
Lentelių sąrašas	4
Anotacija	5
Summary	6
Įvadas	7
1. Web – draudimo sistemos apibrėžimas	9
1.1. Draudimo verslo apibūdinimas	9
1.2. Rizikos valdymas	11
1.3. Pardavimų valdymas ir planavimas	11
1.4. Duomenų apsauga ir auditas	12
1.5. Administravimas	12
1.6. Apskaita	13
1.7. Produktai	13
2. Egzistuojančių algoritmų ir sistemų analizė	14
2.1. Funkcionalumo įvertinimas	14
2.2. Technologinių sprendimų įvertinimas	15
2.3. Silpnų vietų identifikavimas	16
3. Tikslinės srities nustatymas	17
4. Sukurtų algoritmų ir technologijų aprašymas	18
4.1. RAD (Rapid Application Development) produktų kūrimo technologija	21
4.2. Pakeitimų valdymas	23
4.3. Transakcijų valdymo mechanizmas	23
4.4. Delninio kompiuterio interfeisinis priedas	24
4.5. Integracija su išoriniais tiekėjais	24
4.6. Poliso išdavimo procesas	25
4.7. API ir DB ypatybės	27
4.8. Atviras interfeisas į verslo sistemą (pernešamumas)	28
5. Realizavimo ypatybės	30
6. Sistemos įvertinimas	33
Išvados	36
Literatūros sąrašas	38
Priedas Nr.1	41
Priedas Nr.2	42
Priedas Nr.3	46

Paveikslukų sąrašas

Pav. 1 „Draudimo rinka“	14
Pav. 2 „WDA sistemos infrastruktūra“	19
Pav. 3 „XSQL –XSL sąsaja“	20
Pav. 4 „Produktų dizaineris“	22
Pav. 5 „Produktų dizainerio sukurta web forma“	22
Pav. 6 „WDA API architektūra“	28
Pav. 7 „WDA sistemos architektūra“	32
Pav. 8 „Pradinis poliso įvedimas“	42
Pav. 9 „Pakartotinas spausdinimas“	42
Pav. 10 „Pradinis poliso duomenų įvedimas“	43
Pav. 11 „Dokumentų spausdinimas/poliso patvirtinimas“	44
Pav. 12 „Blanko tikrinimas“	45
Pav. 13 „Siūlomas automatinis procesas blankų priskyrimo korekcijai“	45

Lentelių sąrašas

Lentelė 1 „Funkcionalumo palyginimas“	15
Lentelė 2 „Pasirinktos technologijos“	15
Lentelė 3 „3-jų sistemų funkcionalumo palyginimas“	33
Lentelė 4 „Web sistemų našumo palyginimas“	34
Lentelė 5 „Sistemų funkcionalumo palyginimas“	41

Anotacija

Tiriamasis darbas skirtas web draudimo informacinės sistemos modelio analizei ir kūrimui. Pagrindinis darbo tikslas sukonstruoti analitinį draudimo informacinės sistemos modelį, kuris atitiktų kiek įmanoma daugiau verslo sistemos logiką, bei turėtų internetinį interfeisą, kuris leistų vartotojui dirbti nepriklausomai nuo jo buvimo vietos ir laiko. Darbe nagrinėjama ne tik techninė dalis, t.y. kaip sukurti sistemą, bet ir projektinė modelio dalis, t.y. kokia sistema turi būti, tam kad draudimo bendrovė galėtų tvarkingai funkcionuoti. Tiriamajam darbui buvo pasirinktas indukcinis tyrimo metodas, t.y. autoriai analizavo egzistuojančias draudimo informacinių sistemų modelius, bei pritaikė savo patyrimą ir žinias, kad sukurti naują web draudimo informacinę sistemą. Šio tiriamojo darbo rezultatas yra labai naudingas draudimo bendrovėms, kurios jau naudoja sukurtą web sistemą. Ši sistema palaiko beveik visas verslo logikos ypatybes, bei leidžia automatizuoti dalį rankinių procesų. Pagrindinis sukurto modelio privalumas yra sistemos pernešamumo savybė, t.y. draudimo sistemos modelis su nedidelėm išlygom gali būti pritaikytas bet kokiai draudimo bendrovei. Taip pat sukurti greito sistemos vystymo metodai supaprastina produktų kūrimo procesą, bei sumažina sistemos apkrovimą, kas leidžia minimizuoti kompanijos išteklius.

Summary

Thesis title: Web insurance systems design and development.

The key concept of the research proposed by the paper is to analyze the processes of Web insurance information system design and development. To put it in some more simplified words the main objective of the research is to develop analytical insurance information system model, which would confront at the highest possible level, to the business logics, moreover would have Internet interface, which would provide users with favorable working conditions independently on the time and place.

The purpose of the study is multipartite:

- To analyze existing insurance information systems;
- To provide Web insurance information system design mechanism;
- To develop open interface algorithms;
- To improve business processes over their execution time and reliability;
- To construct product manager;
- To evaluate and compare created system with competitor' products.

The research paper analyzes not only technical, but theoretical issues as well. It describes the main aspect of the “ideal” insurance information system that any developer should implicate in the design process. Due to inductive nature of the research existing insurance systems were analyzed and personal experience was included in the new web-based insurance system creation process.

The results provided by this research paper are very useful and important for insurance companies. Moreover some of them already use the newly created product. The main odds of developed model is its portability, meaning that due to the open application interface it could be easily transferred from one company to another as a supplementary tool for the sand alone application. Thereafter developed and implemented rapid application development tools minimize companies' expenditures according to the working time and competences.

Ivadas

Dabartinė Interneto plėtra ir prieinamumas sudaro ir netgi verčia įmones plėstis į web orientuotus pardavimus, kas sudaro galimybes užkariauti naujas rinkas ir sumažinti administravimo kaštus.

E-prekyba egzistuoja jau seniai ir daug autorių nagrinėjo elektorinių parduotuvių sukūrimo įrankius, algoritmus, bei jų veikimo procesus ir principus. Todėl mes savo darbe nagrinėsime ne tiesioginius elektroninių pardavimų kanalus (t.y. savitarnos portalus), o fundamentalios verslo sistemos logikos pernešimą į web erdvę. Tokiu būdu informacinė sistema bus optimaliai naudojama, nepriklausys nuo laiko, vietos, darbuotojų užimtumo, t.y. turės mobilumo savybes, kas ne tik sumažins administracinius kaštus, bet ir padidins pardavimų apyvartą. Be to tokia informacinė sistema galės papildyti pagrindinės verslo sistemos funkcionalumą. Šiam tikslui pasiekti autoriai pasirinko analizuoti draudimo sferą (šiam darbe naudodami terminą „draudimas“ kalbėsime tik apie ne gyvybės draudimo rūšys), kadangi turi patirties ir žinių apie veiklos ypatybes, informacinių sistemų pranašumus ir trūkumus, domisi naujomis interneto technologijomis, bei gali pasiūlyti naujus sprendimų variantus. Taigi šis darbas skirtas web – verslo draudimo informacinių sistemų kūrimui, vystymui, algoritmų projektavimui bei technologijų analizei.

Šio darbo pagrindiniai tikslai yra:

- Atlikti turimų draudimo informacinių sistemų analizę;
- Išvystyti Web verslo sistemos projektavimo mechanizmą;
- Sukurti atviros sąsajos algoritmus;
- Pasiūlyti egzistuojančių procesų pagerinimą, patikimumo bei laiko aspektais;
- Sukurti produktų valdymo įrankį;
- Gautų rezultatų įvertinimas;
- Turimos sistemos bei technologijų palyginimas su jau egzistuojančiomis.

Kadangi teorinių šaltinių yra labai mažai, tyrimas bus vykdomas indukciniu būdu, t.y. iš praktinių žinių, pastebėjimų ir atliktų testų prieisime apibendrintas teorines išvadas. Tyrimas bus vykdomas keliais žingsniais. Pirmiausia autoriai nagrinės egzistuojančias web draudimo informacinės sistemas, jų privalumus bei trūkumus, išvardins naudojamą technologijas, bei sukurs „Tobulos“ web – draudimo informacinės sistemos koncepcinį modelį. Sekančiame žingsnyje bus kuriami procesai, algoritmai bei technologiniai sprendimai koncepcinio modelio įgyvendinimui. Kadangi pagrindinis tikslas pasiūlyti atviros sąsajos sistemą, bus įvertintos skirtingų duomenų bazių, programinės, techninės įrangos galimybės. Bus analizuojamos XML, XSQL, bei XSL technologijų naudojimo suteikiami privalumai. Vėliau bus pateikti sukurtos sistemos pagrindiniai algoritmai bei procesų įvertinimai.

Šio tiriamojo darbo rezultatas – sukurti, nepriklausomą web sistemą, kuri tiktų bet kokiai draudimo bendrovei, turėtų produktų valdymo mechanizmą, bei sąsajas su išoriniais duomenų tiekėjais. Kalbant apie draudimo bendroves, tam, kad ne klaidinti skaitytojo norime patikslinti, kad sukurta sistema, t.y. verslo sistemos logika atitiks bet kurios draudimo bendrovės lūkesčius, bei veikimo principas ir naudojamos technologijos nepriklausys nuo konkrečios bendrovės, bet pati realizacija, t.y. Application Programming Interface(API) gali būti koreguojama priklausomai nuo pagrindinės verslo sistemos architektūros. Be to sukurtai sistemai bus adaptuoti keli sistemos išplėtimai, kurie padidins sistemos pasiekiamumą ir mobilumą, t.y. sistema bus lengvai pritaikoma ir naudojama stand-alone aplikacijose, delniniuose kompiuteriuose bei mobiliuose telefonuose.

1. Web – draudimo sistemos apibrėžimas

Prieš pateikiant skaitytojui IT draudimo sistemos apibrėžimą, norėtume trumpai aptarti draudimo veiklos esmę ir ypatumus. Draudimo verslas yra savotiškas fenomenas visoje ekonominėje rinkoje, todėl pradėsime nuo bazinio draudimo verslo apibūdinimo.

1.1. Draudimo verslo apibūdinimas

Draudimo paslaugų esmė - sukauptų lėšų, gautų iš draudimo įmokų, paskirstymas, atlyginant galimus nuostolius skirtingu laiku ir įvairiose vietose. Draudimo fondas surenkamas per specialias organizacijas - draudimo bendroves. Jis naudojamas draudėjų patirtiems nuostoliams dėl stichinių nelaimių, nelaimingų atsitikimų ar kitų įvykių, kurie numatomi draudimo sutartyse, kompensuoti.

Pagrindiniai draudimo bruožai:

- Draudimas - tai rizikos perdavimas draudikui;
- Draudimas - tai bendras fondas, todėl nuostoliai paskirstomi visoje draudimo grupėje;
- Riziką galima mažinti remiantis draudimo statistika ir naudojantis matematiniais metodais - draudikas gali prognozuoti būsimus nuostolius;
- Draudimo veikloje perduodamos tik rizikos finansinės pasekmės (nuostoliai).

Draudimo verslo specifika, funkcijos ir vieta šiuolaikinėje ūkio sistemoje

Draudimo rinka - tai ypatingų santykių sfera, susijusi su specifinės prekės – draudimo paslaugos – pirkimu ir pardavimu. Draudimo paslauga yra specifinė todėl, kad ji vienu metu yra ir vartotojiška, ir finansinė. Vartotojiška draudimo paslaugos savybė yra ta, kad ją draudėjai perka negamybiniam naudojimui. Draudimas, kaip finansinė paslauga, ypatingas tuo, kad yra perskirstomi finansiniai resursai. Viena iš draudimo paslaugų savybių yra ta, kad ji parduodama anksčiau, negu pateikiama, t.y. klientas (draudėjas) mainais į sumokėtą draudimo įmoką įgyja draudimo apsaugą, bet jos materialiąją išraišką - draudimo išmoką jis gaus tik po draudiminio įvykio. Galima išskirti tokias pagrindines draudimo paslaugų funkcijas:

- Suteikti fiziniams ir juridiniams asmenims finansinę paramą atsitikus nelaimei, apsaugoti nuo netikėtų finansinių sunkumų;
- Apsaugoti rinkos dalyvius nuo verslo nestabilumo, finansinės padėties svyravimų;
- Padėti konstruktyviai ir sistemingai siekti finansinių tikslų;
- Draudimas atlieka kapitalo ir pajamų išsaugojimo funkciją. Pajamų išsaugojimas yra darbo vietų išsaugojimo pasekmė;

- Kreditų išsaugojimo funkcija. Paskolos taptų rizikingesnės bei jų išdavimas sumažėtų, jei neegzistotų nekilnojamojo turto, esančio kredito gražinimo garantu, draudiminė apsauga;
- Netiesioginė funkcija - papildomų darbo vietų sukūrimas.

Pasaulio draudimo rinka, kuri yra viena iš finansų rinkos, egzistuojančios kaip savarankiškas ekonomikos fenomenas ir besiformuojančios kaip nacionalinių rinkų visuma, sudedamųjų dalių, lemiamą globalizacijos proceso permąnų patiria daugelį pasikeitimų.

Šiuolaikinėje pasaulinėje ekonomikoje draudimo vaidmuo labai didelis. Galima išskirti tokius pagrindinius požymius:

- draudimas yra viena iš pagrindinių privačių "rizikos valdymo" koncepcijų rizikos valdymo formų;
- draudimas ir perdraudimas, užtikrinant globalių rizikų perskirstymą, draudiminio įvykio atveju užtikrina draudimo kompensacijų, kartais dešimčių milijardų eurų, išmokėjimą;
- draudikai yra stambiausi investuotojai tiek nacionalinių ekonomikų, tiek tarptautiniu mastu;
- draudikai papildo valstybinę draudimo ir aprūpinimo sistemą ir iš dalies ją keičia (savo investicijomis), užtikrina būtiną gyventojų socialinės apsaugos lygį.

Web – draudimo sistemos apibrėžimas

Manome, kad dabar skaitytojas supranta kas yra draudimo veikla ir šios veiklos specifika, todėl pateiksime supaprastintą draudimo informacinės sistemos apibrėžimą.

Informacinė draudimo sistema turi atitikti visas aukščiau išvardintas veiklos savybes. Tai nėra įprastas programinės įrangos paketas, kuris leidžia išsaugoti duomenų bazėje draudimo poliso pardavimo faktą, svarbiausia, kad informacinė sistema palaikytų veiklos logiką, galėtų automatizuoti kuo daugiau procesų, bei sektų draudimo poliso pasikeitimus ir istoriją.

Pagrindinės draudimo informacinės sistemos ypatybės yra:

- padėti valdyti riziką, bei akumuliuoti sukauptus duomenis apie klientus bei įvykius;
- turėti pardavimų valdymo ir planavimo mechanizmą;
- užtikrinti duomenų saugumą, kadangi pagrindinė vertė kuria disponuoja draudikas yra kliento duomenys;
- sutarčių, bei žalų administravimo galimybės;
- draudimo produktų kūrimo sąsaja;
- finansinės apskaitos administravimas.

Trumpai aptarsime kiekvieną iš išvardintų savybių iš techninės pusės, kadangi būtent jos padeda vartotojui būti pranašesniai rinkoje, t.y. nustatysime koks funkcionalumas yra būtinas

draudimo informacinės sistemos kiekvienam iš nurodytu modulių. O kalbant apie Web draudimo informacinę sistemą, norime pabrėžti, kad tokiu būdu vartotojas turės daugiau galimybių, kurios suteiks jam tik papildomą pranašumą.

1.2. Rizikos valdymas

Rizikos valdymas yra vienas iš pagrindinių draudimo bendrovės uždavinių, kadangi draudikas prisiima apdraustojo riziką ir garantuoja išmokėjimą įvykiui atsitikus. Ne visos dabartinės draudimo informacinės sistemos turi tokį, autorių nuomone vieną iš kritinių, funkcionalumą:

- Kelių lygių autorizacijos sistema, leidžianti paskirstyti atsakomybę už sutarties autorizaciją pagal rizikos vertintojų kompetencijų ribas;
- Draudžiamų asmenų/objektų segmentaciją pagal rizikos vertintojų nustatytus kriterijus. Segmentaciją galima padaryti priklausančią nuo produkto ir kintamą laike;
- Automatinis rizikingų klientų (juodasis sąrašas) sutarčių blokavimas iki vindikacijos skyriaus išvadų;
- Sutarties tikrinimas pagal specifinės produktui taisyklės, techniškai kalbant validavimo taisyklės;
- Rizikos padidėjimo atveju performintos sutartys sustabdomos iki rizikos vertintojų išvados. Sustabdymo metu galioja paskutinė autorizuota sutarties versija.
- Privalomas sutarties autorizavimo priežasčių fiksavimas;
- Vieningas draudžiamų objektų/asmenų registras, su objekto istorijos fiksavimu.
- Objektų identifikavimo taisyklės nustatymas pagal objektų klases (galimas tikrinimas išorines sistemose – Regitra, Kadastras, GIS):
 - Transportas – valstybinis ir/arba registracijos numeris;
 - Pastatai – kadastro numeris, galimybė nurodyti globalines koordinatės, tikslu nustatyti priklausomybę rizikingam pagal produkto segmentus rajonui.

1.3. Pardavimų valdymas ir planavimas

Kadangi draudimo bendrovė yra finansinė institucija, bei jos pelnas priklauso nuo pardavimų apimties, kyla natūralus noras valdyti padavimo kanalus, turėti įrankius kurie padės draudikui teisingai įvertinti klientą, pasiūlyti jam patrauklų tarifą, ar tiesiog pasiūlyti klientui apsidrausti pačiam, be agento įsikišimo į šį procesą. Taigi svarbiausi, autorių nuomone, yra šie kriterijai:

- Kvotų posistemė;
- Tikslinių nuolaidų posistemė;
- Klientui pageidaujant, automatinis sutarties atnaujinimas;
- Lanksti bonus/malus sistema, su galimybe tikrinti kaip sutarties istoriją, taip ir visą kliento portfeli;

- Marketinginis modulis:
 - Trumpalaikis įkainių, nuolaidų, sąlygų keitimas pardavimo akcijų metu, nekeičiant produktų realizacijos kodų;
 - Marketinginės kompanijos paruošimo išlaidų fiksavimas – planuojamos, faktinės;
 - Tiesioginis polisų sąryšis su pardavimo akcija, leidžia tiksliai atlikti marketinginės kompanijos efektyvumo analizę: tiek pardavimų, tiek nuostolingumo prasme;
- Savitarnos portalas. Per bankinę E-sistemą klientas gali sudaryti sutartį pagal iš anksto paruoštą produktų sąrašą. Tokia sutartis gali būti siunčiama autorizuoti. Apie tai klientui pranešama iš karto. Informacija apie rizikos vertintojo sprendimą kartu su siūlymu sumokėti už sutartį, klientas gauna elektroniniu paštu. Apmokėjimas vyksta per bankinę E-sistemą, polisas spausdinamas spaustuvėje ir siunčiamas paštu klientui arba klientas turi galimybę atspausdinti sutartį pats, priklausomai nuo įstatyminės bazės ir pačio kliento pageidavimų. Komisinis mokestis už tokias sutartis neturi būti mokamas arba bus labai mažas.

1.4. Duomenų apsauga ir auditas

Kaip jau minėjome pagrindinė vertybė kuria disponuoja draudimo bendrovė yra kliento - asmens duomenys, be to, dėl išorinių pardavėjų (brokerių) paslaugų naudojimo, turimi duomenys negali būti laisvai prienami kiekvienam sistemos vartotojui. Kaip rodo praktika ypač pastaruoju aspektu daug sistemų neturi tinkamo sprendimo, pavyzdžiui, tokia galinga draudimo informacinė sistema, kaip „TIA“ kiekvienam prisijungusiam vartotojui suteikiama galimybė matyti duomenis apie visus klientus, bei visus parduotus draudiko polisus. Mūsų manymų labai svarbu apriboti peržiūros teises vartotojams, pagal jų kompetenciją:

- Vartotojų profailų sistema, ribojanti arba praplečianti jų priėjimą prie duomenų;
- Vartotojų visų atliekamų veiksmų fiksavimas (vartotojas, laikas, dokumentas, darbo stoties IP adresas), įskaitant duomenų peržiūrą.

1.5. Administravimas

Draudiko bendrovės santykiai su klientu, skirtingai nuo daugumos paslaugų pardavėjų tik prasideda pardavimo metu, todėl labai svarbu, kad draudimo informacinė sistema palaikytų sutarties administravimą jos galiojimo metu. Dabartiniai faktai rodo, kad ne visos sistemos siūlo tokį funkcionalumą.

- Privalomas nutraukimo priežasčių fiksavimas;
- Privalomas performinimo priežasčių fiksavimas;

- Klasifikuojamos sutarties keitimo/nutraukimo priežastis – bendri, ir specifiniai produktai.

1.6. Apskaita

Kaip ir bet kuri finansinė institucija draudimo bendrovė privalo apskaityti pinigus. Žinoma, viena iš galimybių, yra tvarkyti finansus kokioje nors specialioje finansinėje sistemoje, pvz. Oracle Financials arba Navision. Kita galimybė – apskaityti poliso įmokas pačioje draudimo sistemoje. Pastarasis atvejis yra patogesnis poliso administravimo klausimais, būtent todėl mes manome, kad draudimo IS būtina turėti:

- Apskaita pagal sutartį, produktą arba draudžiamą objektą, priklausomai nuo pardavimo vieneto logikos;
- Apskaita pagal kalendorines dienas;
- Mokėjimo grafikas siūlomas pagal pardavimo vieneto šabloną, klientui pageidaujant gali būti pakeistas į kitą. Produkto logiką gali riboti mokėjimo grafiko keitimus;
- Ilgalaikės sutartys.

1.7. Produktai

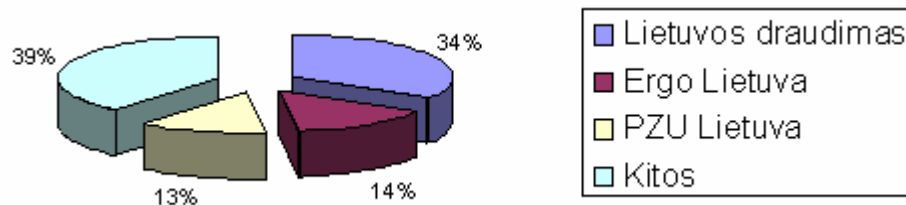
Produktas tai yra draudimo kompanijos paslauga, kurią perka klientas. Vienas iš svarbiausių kriterijų renkantis draudiką, šiuo metu yra kaina, siūlomas apsaugos lygis ir aptarnavimo greitis. Dėl šios priežasties kainos konkurencingumą galėtų palaikyti sudėtiniai produktai, o duomenų išoriniai tiekėjai pagreitintų pardavimo procesą.

- Comprehensive polisai (pvz. šeimos turtas – būstas, transportas, namų turtas – viskas vienoje sutartyje);
- Produkto lygyje numatyta galimybė imti duomenis iš išorinių sistemų – Motorinio biuro, Regitros ir t.t..

2. Egzistuojančių algoritmų ir sistemų analizė

Šioje dalyje mes aptarsime egzistuojančias draudimo informacinės sistemas, pabrėšime jų privalumus ir trūkumus. Kaip jau minėjome abu autoriai dirba draudimo srityje ir yra sukaupę nemažai patirties šioje sferoje, būtent dėl šios priežasties jie pasirinko tokią tiriamojo darbo temą, kadangi mato ką galima pasiūlyti ir nori sugalvoti optimalius sprendimus savo pasiūlymams įgyvendinti. Tam, kad pagrįsti savo idėjų svarbą ir aktualumą pateiksime šiuo metu naudojamu Lietuvos 3 didžiausių draudikų informacinių sistemų analizę. Tai yra teorinis tyrimo pagrindas, kadangi pagrindiniai duomenų šaltiniai buvo būtent techninės šių sistemų specifikacijos.

Kaip žinoma šiuo metu Lietuvoje didžiausios draudimo bendrovės yra: UAB DK „Lietuvos draudimas“, UAB DK „Ergo Lietuva“, UAB DK „PZU Lietuva“. Pirmoji ir didžiausia draudimo bendrovė naudoja danišką draudimo informacinę sistemą – „TIA“, o „Ergo Lietuva“ ir „PZU Lietuva“ yra maždaug vienodo dydžio, dalijasi antrą ir trečią vietas rinkoje ir naudojami lietuviškos kompanijos siūloma informacinė sistema – „DIS“. Nors abi sistemos skirtos draudimo kompanijoms, jos yra labai skirtingos savo funkcionalumo prasme, pasirinktais technologiniais sprendimais. Todėl sekančioje dalyje pateiksime išsamų šių sistemų įvertinimą.



Pav. 1 „Draudimo rinka“

2.1. Funkcionalumo įvertinimas

Istoriškai taip susiklostė, kad abiejų sistemų autoriai yra buvę draudimo kompanijų Informacinių Technologijų departamentų darbuotojai, kurie dėl tam tikrų priežasčių susivienijo į atskiras kompanijas, kurios šiuo metu pardavinėja savo sukurtus produktus „TIA“ (The Insurance Application) ir „DIS“ (Draudimo Informacinė Sistema). Faktas, kad sistemų architektai ir kūrėjai išmanė draudimo sferą, suteikia vilčių, kad aukščiau aprašyti draudimo informacinės sistemos reikalavimai turėtų būti išpildyti. Bet ar taip yra iš tikrųjų?

Tam, kad atsakyti į šį klausimą pateiksime skaitytojui sistemų funkcionalumo palyginamąją analizę. Analizė buvo atliekama pagal turimas sistemų specifikacijas, bei atsižvelgiant į konsultacijas su sistemų platintojais.

Funkcionalumas	TIA	DIS
<i>Autorizacijos modulis</i>	+	-
<i>Validavimo modulis</i>	+	+
<i>Kvotų posistemė</i>	-	-
<i>Tarifikacijos modulis</i>	+/-	+/-
<i>Sutarties administravimo galimybės</i>	+	-

Lentelė 1 „Funkcionalumo palyginimas“

Pilna funkcionalumo palyginimo suvestinė pateikta Priede Nr.1

Kaip matome iš pateiktų duomenų abi sistemos turi galimybę patikrinti įvestus duomenis, bei tarifavimo modulius. Prie paskutiniojo pažymėjimas (+/-) reiškia, kad abi sistemos sugeba apskaičiuoti įmokas automatiškai, bet tarifų pakeitimams būtinas sistemos keitimas, t.y. programavimas. Kitaip tariant verslo žmonės patys negali koreguoti tarifų posistemės. „TIA“ draudimo aplikacija turi savo autorizacijos modulį (vieno lygio autorizacijos), bei palaiko sutarties administravimą. Pastarasis aspektas yra pagrindinis „DIS“ trūkumas, kadangi juridinio sektoriaus sutartys yra keičiamos ne vieną kartą metų bėgyje. Taigi nesugebėjimas įvertinti, bei išsaugoti sutarties pakeitimus prie pradinės sutarties duomenų yra tikrai didelė problema. „DIS“ siūlomas sprendimas yra sutarties nutraukimas bei naujos sutarties (su papildymais) registravimas sistemoje, be to sąryšio tarp sukurtų sutarčių nėra, bet tai jau techninė sistemos pusė.

2.2. Technologinių sprendimų įvertinimas

Sistemų technologiniai sprendimai irgi skiriasi. Čia žinoma sunku pasakyti, koks technologijų pasirinkimas yra pranašesnis, svarbiausia, kad sistema veiktų greitai, patikimai ir atitiktų verslo keliamus reikalavimus.

Technologijos	TIA	DIS
<i>Duomenų bazė</i>	Oracle	Oracle
<i>Interfeisai</i>	Oracle Forms + Java	Gupta
<i>Sistemos logika</i>	Serverio dalis	Klientinė dalis
<i>Veikimo greitis (duomenų apsikeitimas)</i>	0,01 sek.	0,01 sek.
<i>Transakcijų skaičius (TPVCA poliso įvedimo metu)</i>	~20	~100

Lentelė 2 „Pasirinktos technologijos“

Kaip matome abi sistemos veikia su Oracle DB šiuo metu 9 versija. Interfeisų konstravimui pasirinktos skirtingos technologijos, kuris iš pasiūlytų sprendimų yra geresnis, tikriausiai yra skonio reikalas, bet Oracle Forms yra plačiau žinomas, kita vertus jis yra ir žymiai brangesnis.

Kertinis akmuo čia būtent sistemos logikos vietos pasirinkimas. Ar tai turi būti darbo stotis (Server – side) ar klientinė dalis (client – side)? Kaip matome sistemų architektai pasirinko skirtingas strategijas. TIA – sistema visą sistemos logiką perkėlė į darbo stotį, kaip Oracle Stored Procedures, o DIS – pasirinko klientinę dalį. Šiuo atžvilgių lengva pasakyti, kad DIS pasirinkimas smarkiai įtakoja jo veikimo greitį, t.y. sistemos paleidimas gali užtrukti iki kelių minučių ir tai negatyviai vertinama vartotojų tarpe. Žinoma galima pagalvoti apie nuotolinį prisijungimą per terminalinę darbo stotį, kuris tikrai pagerintų sistemos darbo laiką. Bet čia vėl susiduriame su išlaidomis, pagal dabartinius duomenis draudimo bendrovė „Ergo Lietuva“ turi apie 10 terminalinių darbo stočių, tam, kad užtikrinti pakankamą sistemos veikimo greitį. Tuo tarpu draudimo bendrovė „Lietuvos draudimas“ turi tik 4, o dirbančių sistemoje žmonių skaičius yra ženkliai didesnis, negu pas konkurentus.

2.3. Silpnų vietų identifikavimas

Šioje dalyje tiesiog apibendrinsime aukščiau atliktą analizę. „TIA“ ir „DIS“ draudimo informacinės sistemos yra visiškai skirtingos, nors ir skirtos tam pačiam sektoriui, vienodiems poreikiams patenkinti. Šio tiriamojo darbo tikslas nėra nustatyti kuri iš sistemų yra geresnė ar blogesnė, autorių nuomone tai konkrečių vartotojų prerogatyva. Tiesiog norime identifikuoti silpnąsias šiuolaikinių sistemų vietas ir pasiūlyti savo sprendimo variantus, kaip galima pasiekti norimo funkcionalumo, veikimo greičio ar kitokių savybių realizavimą. Taigi norime išskirti pagrindinius egzistuojančių sistemų trūkumus:

- Pilnaverčiam rizikos valdymo moduliui būtinas papildymas kelių lygių autorizacijos posisteme, t.y. paskirstyti atsakomybę už rizikos prisiėmimą pagal rizikos vertintojų kompetencijas;
- Nepalaikoma klientų segmentaciją pagal jų rizikingumo lygį;
- Būtinas sutarties administravimas, palaikant sąryšį tarp naujai sudarytos (pakeistos) sutarties ir pradinės sutarties nėra įdiegtas visose sistemose;
- Patenkinamo veikimo greičio užtikrinimas (iki 5 minučių vieno poliso įvedimui);
- Nėra transakcijų valdymo mechanizmo;
- Nėra standartinių kvotų bei tarifų valdymo mechanizmų;
- Nėra vartotojo veiksmų sekimo galimybių.

3. Tikslinės srities nustatymas

Šio tiriamojo darbo tikslas sukurti Web – draudimo informacinę sistemą, kurią būtų galima naudoti kaip savarankišką aplikaciją arba kaip praplėtimą su jau egzistuojančia sistema, pavyzdžiui „TIA“ arba „DIS“. Autoriai nori išvystyti draudimo informacinės sistemos projektavimo mechanizmą, t.y. sukurti metodologiją kaip ir į ką reikia atkreipti dėmesį projektuojant tokio tipo sistemas. Pasiūlyti atviros sąsajos algoritmus, kurie leistų be žymesnių pakeitimų pereiti nuo vienos pagrindinės aplikacijos prie kitos. Žinoma tai reikalauja tam tikro sistemos perdarymo, mūsų tikslas yra, sukurti tokią schemą, kad tokių pakeitimų kiekis būtų minimalus. Kaip jau ne kartą sakėme, labai svarbus yra sistemos veikimo greitis, t.y. sistema turi atitikti vartotojų keliamus reikalavimus. Vienos transakcijos laikas neturi viršyti 0,01 sek., techniniai naudojamo kompiuterio reikalavimai turi būti minimalūs, TPVCA poliso registravimo laikas neturi viršyti 3 minučių. Sukurta sistema, neturi būti tiesiog poliso įvedimo mechanizmas, tai pilna draudimo verslo sistema, kuri leidžia automatizuoti kiek įmanoma daugiau procesų, palaiko sutarties administravimą jos egzistavimo laikotarpiu, taip pat suteikia papildomą funkcionalumą verslo poreikiams įgyvendinti, minimizuoja įvedamų duomenų kiekį, t.y. visa įmanoma informacija turi būti imama iš išorinių duomenų tiekėjų, leidžia vartotojams dirbti bet kuriuo paros metu, nepriklausomai nuo jų buvimo vietos. Be to pasirinkti technologiniai sprendimai yra labai paprasti, t.y. nereikalaujantis didelių finansinių išlaidų sistemos diegimui. Mūsų pagrindinis tikslas pilnai realizuoti mūsų apibrėžtą web – draudimo informacinę sistemą, bei pasiūlyti kai kuriuos praplėtimus prie realizuotos sistemos.

Interneto technologijos plinta labai sparčiai ir jų populiarumas tarp vartotojų auga kiekvienais metais. Nors mūsų darbe mes kalbame apie draudimo sistemą skirtą draudimo kompanijos darbuotojams ir išoriniams pardavėjams, reikia pabrėžti tai, kad auga ir procentas žmonių kurie norėtų įsigyti draudimo polisą internetu, be jokio trečios šalies įsikišimo. „PZU Lietuva“ atliktų apklausų statistiniai duomenys rodo, kad nuo 2005 iki 2006 metų suinteresuotų žmonių skaičius išaugo netgi 10%, nuo 3% iki 13%. Autorių manymu tai yra rimtas rodiklis ir draudimo bendrovės turi pasiruošti šiam žingsniui. Toks mūsų atliktas tiriamasis darbas yra geras pagrindas savitarnos draudimo portalo kūrimui, kadangi technologiniai sprendimai bei sukurti algoritmai ir procesai yra universalūs, t.y. savitarnos portalas yra tiesiog siūlomos sistemos praplėtimas, o ne nauja sistema.

4. Sukurtų algoritmų ir technologijų aprašymas

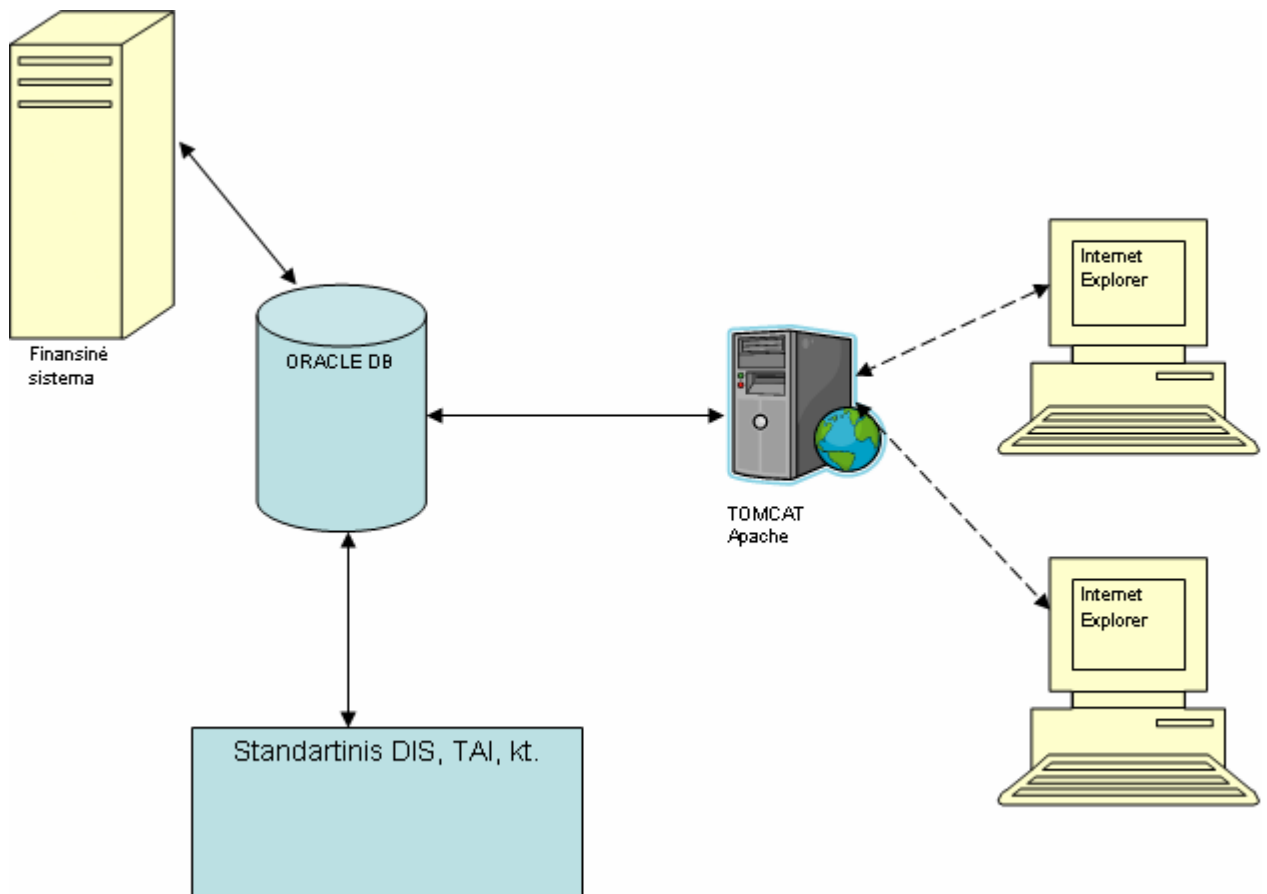
Šiame skyriuje aprašysime sukurtą mechanizmą, bei pačią programinę įrangą kurią vadinsime WDA –Web Draudimo Aplikacija. Be to aprašysime sukurtus algoritmus/ mechanizmus, kurie padėjo mums pasiekti užsibrėžtus tikslus.

Pirmiausia aptarsime pasirinktus techninius sistemos parametrus, kadangi būtent jie lemia sistemos kainą, t.y. ar yra reikalingos kokios nors specialios ir brangios tarnybinės stotys ar programinė įranga?

Iš tikrųjų pasirinktas technologinis sprendimas yra pakankamai pigus ir paprastas. Kaip ir aukščiau aprašytos draudimo sistemos, Web – draudimo informacinei sistemai, WDA, mes pasirinkome Oracle duomenų bazės platformą. Šį pasirinkimą lėmė keli faktoriai:

- Oracle yra stabiliai veikianti ir plačiai naudojama duomenų bazė;
- Dauguma siūlomų sistemų („TIA“ , „DIS“) naudoja šia duomenų bazę, kadangi Web sistema gali būti naudojama kaip pagrindinės sistemos papildymas, kitaip sakant web – moduliui logiška pasirinkti tą pačią duomenų bazę;
- Oracle pradėdant nuo 9 versijos palaiko darbą su XML formatu, kuris yra plačiai naudojamas sukurtuose algoritmuose.

Interfeisai yra aprašomi naudojant paprasčiausius JavaScript, XSL, Css, HTML ir Java servletus. Web serveriui pasirinktas Tomcat, kuris su XSQL servleto pagalba bendrauja su duomenų baze. Šioje vietoje vietoj Tomcat serverio galima pasirinkti WebSphere siūloma sprendimą, kuris gali būti šiek tiek stabilesnis, bet pradėdant nuo Tomcat 5 versijos yra palaikomas blokinis (cluster) mechanizmas, kuris sėkmingai veikia ir leidžia vienu metu dirbti ~1000 vartotojų. Taigi yra akivaizdu, kad toks techninės bei programinės įrangos derinys nereikalauja didelių investicijų, bei pakankamai stabiliai užtikrina sistemos darbo palaikymą.



Pav. 2 „WDA sistemos infrastruktūra“

Sistemos projektavimas – tai yra vienas iš svarbiausių procesų, sistemos kūrimo metu. WDA sistemos logika perkelta į duomenų bazę, t.y. serverio dalį, analogiškai kaip ir „TIA“ sistemoje tam naudojame Oracle procedūras (stored procedures). Bet nuo pat sistemos projektavimo pradžios norėjome užtikrinti saugų ir patikimą transakcijų valdymo mechanizmą, bei audituoti visus vartotojų veiksmus, t.y. kas, kada, iš kur buvo prisijungęs, kiek laiko dirbo su sistema, kokias klaidas padarė it t.t.. Be to atviro interfeiso sąsajos idėja reikalavo gerai suprojektuotų sprendimų.

Iš karto norime pabrėžti, kad WDA – dabartinė realizacija yra naudojama kartu su „DIS“ sistema, todėl reikėjo atsižvelgti į tam tikrus pagrindinės sistemos niuansus.

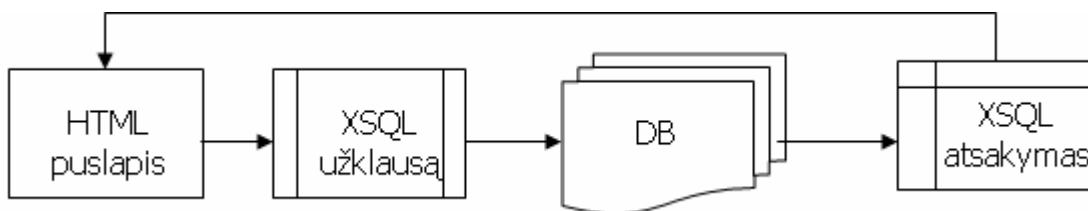
Taigi pradėsime nuo WDA vidinių procedūrų aprašymų bei duomenų bazės sukonstravimo. Kaip jau minėjome WDA naudojame XML duomenų formatą, kuris leidžia standartizuoti duomenų apskaitos procesą, t.y. kiekvienam draudimo produktui nereikia aprašyti savo schemas, reikia tik pakeisti konkretaus draudiminio objekto duomenis, o bendri poliso duomenys lieka vienodi visiems produktams. Taip įvedimo formoje visa vartotojo įvesta informacija siunčiama į duomenų bazę XSQL servleto pagalba XML formatu. Duomenų bazėje buvo sukurta lentelė, kurioje saugomi visi polisai įvesti naudojant WDA sistemą būtent XML formatu. Visos šiuo metu autoriams žinomos egzistuojančios sistemos palaiko daugiatransakcinį mechanizmą, t.y.

dažniausiai, duomenų bazės įrašas atsiranda jau tuomet, kai vartotojas, dar net nepradėjęs įvedinėti duomenų ar skaičiuoti įmokos, t.y. jis tik atidarė įvedimo formą. Tokiu būdu duomenų bazėje lieka gan daug „šiuokšlių“, t.y. kiekvienas toks formos atidarymas palieka savo žymą duomenų bazėje, bet nebūtinai formos atidarymas reiškia užbaigtą pardavimo transakciją, t.y. poliso registravimą ir išdavimą. Todėl mes atsižvelgėme į šituos trūkumus, WDA sistemoje duomenų apsikeitimas, t.y. rašymo į DB veiksmas atliekamas tik skaičiuojant ir registruojant polisą. Toliau naudojant mūsų sukurtą ir aprašytą „DIS“ API poliso registravimo metu įvesti duomenys yra replikuojami į „DIS“ sistemą, t.y. užpildomos visos reikalingos lentelės, taip, kad polisą galima būtų matyti iš abiejų sistemų.

Sekantis svarbus aspektas yra auditas, kiekvienas vartotojas prisijungdamas prie WDA sistemos palieka žymą: vartotojo id, paskutinio atlikto veiksmo laiką, IP adresą iš kurio vykdomas prisijungimas bei suteiktas sesijos id. Be to taip pat nutarėme, kad yra svarbu audituoti ir nepavykusius prisijungimus prie sistemos, tokių būdu sumažėja įsilaužimo galimybės, be to jeigu taip ir atsitiktų galėtume atsekti kas ir kaip tai padarė.

Atviro interfeiso mechanizmą užtikrinome, būtent API įvedimo pagalba. Šiuo metu egzistuoja dar vienas internetinis „TIA“ modulis, kuris dirba tiesiai su duomenų baze, bei turi milžinišką transakcijų skaičių vieno poliso įvedimui. Žinodami šį faktą norėjome apeiti šį nepatogumą, bei sukurti globalų produktą, kuriuo galėtu naudotis ne viena Lietuvos respublikos draudimo bendrovė.

Interfeisinė dalis yra gan paprasta, bei nereikalaujanti didelių programavimo įgūdžių, kadangi naudoja paprasčiausias skriptines kalbas ir XML'ą. Įdomesnis momentas čia yra formų išdėstymas, bei XSQL – XSL tarpusavio sąsaja duomenų apsikeitimui. WDA sistemoje atsisakėme pop-up langų, t.y. visas vartotojo poliso įvedimo interfeisas yra sudarytas viename IE naršyklės lange, tai supaprastina vartotojo darbą, bei pagreitina poliso įvedimo procesą. Apsikeitimas su duomenų baze vykdomas naudojantis XSQL – XSL technologijomis. Schematiškai tai atrodo šitaip:



Pav. 3 „XSQL –XSL sąsaja“

Dar vienas papildomas WDA sistemos privalumas yra sukonstruotas poliso įvedimo procesas, kuris gali būti padalintas į dvi pagrindines dalis:

- Pradinis poliso duomenų įvedimas/dokumentų spausdinimas;

- Pakartotinas spausdinimas/dublikatų išdavimas.

Pilną sutarties sudarymo procesą galima rasti priede Nr.2.

Toliau plačiau aprašysime visus sukurtus įrankius.

4.1. RAD (Rapid Application Development) produktų kūrimo technologija

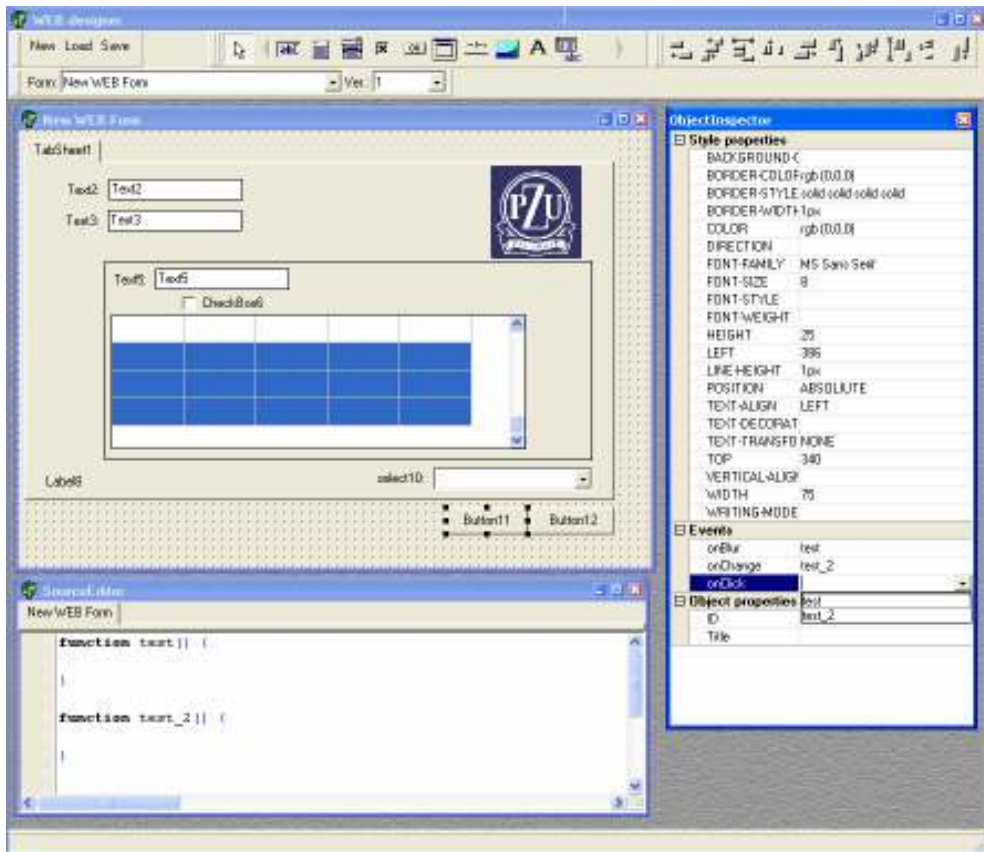
Kadangi draudimo versle labai dažnai atsiranda nauji produktai, bei keičiami jau egzistuojantys, yra labai patogu turėti įrankį kurio pagalba, vartotojo įvedimo forma galima būtų realizuoti tiesiog „drag & drop“ principu, t.y. be papildomo programuotojų įsikišimo į šį procesą. Taip sutaupytime laiką, bei kiekvienas už produktą atsakingas asmuo galėtų pats įgyvendinti savo fantazijas. Taigi nutarėme suteikti tokią galimybę ir sukurti produktų kūrimo dizainerį. Šis sistemos praplėtimas buvo sukurtas su Delphi ir jis interpretuoja nupaišytus laukus, bei formas į HTML priimtina formata, t.y. vartotojas nupaišo tam tikrus laukus, lenteles, formas ar kūnus (frame), tada sukurtas įrankis interpretuoja nurodytus objektus, jų stilius, buvimo vietą (pagal absoliučią poziciją) ir kitus nurodytus atributus ir išsaugo sukurtą failą HTML pavidalu arba užrašo sukurtas įvedimo formas į duomenų bazę. Kadangi sukurtos sistemos įvedimo formos yra dvejų tipų:

- statiškos ir bendros visiems draudimo produktams (asmens ir poliso duomenys);
- dinamiškos (konkreto produkto realizacija).

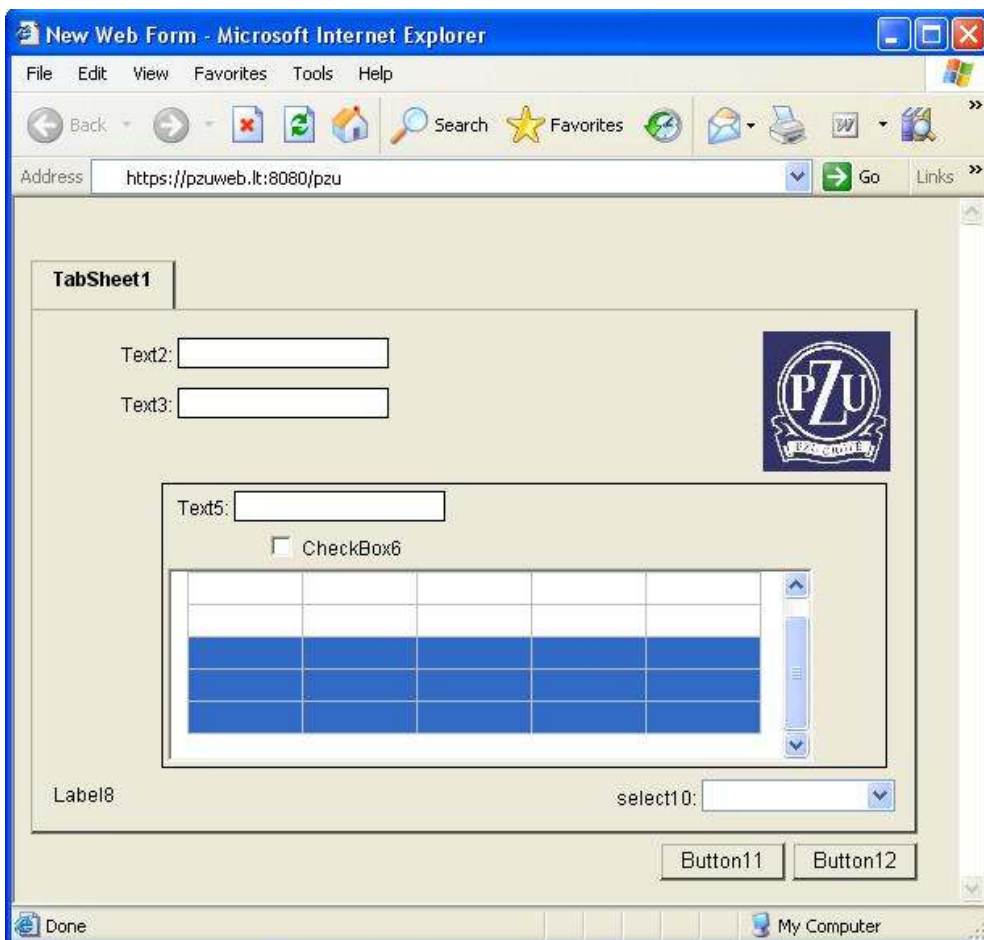
Todėl pirmu atveju siūlome naudoti tiesiog HTML failų sudarymą, o antru dinamiškai apdoroti produkto laukų išdėstymą, t.y. naudotis DB, kadangi tokie objektai yra dažnai keičiami.

Šiuo metu įrankis turi tam tikrus aprašytus laukus bei stilius adaptuotus konkrečiai kompanijai, t.y. jis sėkmingai atlieka savo funkciją produktų kūrime, bet kol kas dar nėra pilnai adaptuotas, bet kokiam (su WDA nesusijusiam) HTML puslapio kūrimui.

Šiuo metu autoriai nagrinėja „XML Data Islands“ privalumus, kurie leidžia supaprastinti sukonstruotų formų išėjties tekstą, bei iš karto operuoja duomenimis, t.y. eliminuoja pradinių duomenų gavimo užklausas, ko pasėkoje padidėja sistemos veikimo greitis. Tyrimas parodė, kad toks duomenų gavimo ir formavimo metodas yra vis daugiau naudojamas tarp įvairių Web aplikacijų kūrėjų.



Pav. 4 „Produktų dizaineris“



Pav. 5 „Produktų dizainerio sukurta web forma“

4.2. Pakeitimų valdymas

Kaip jau ne kartą minėjome draudimo verslas nėra elementarus polisų pardavimas, draudiko ir draudėjo santykiai tik prasideda poliso įsigaliojimo metu, todėl svarbu turėti galimybes administruoti sudarytas sutartis. WDA sistema palaiko šį funkcionalumą, t.y. galima koreguoti egzistuojančios sutarties duomenis, ją išsaugoti bei atspausdinti naujus dokumentus. Duomenų bazės lygyje atsiras naujas įrašas su nuoroda į jos tėvą. Tokiu būdu užtikrinamas pagreitintas administratorių darbas, t.y. kiekvieną kartą nereikia įvedinėti naują sutartį bei nutraukinėti senąją. Be to validavimo bei autorizavimo moduliai gali aprašyti papildomas taisykles koreguoto poliso įvedimui, saugojimui, bei tvirtinimui.

Algoritminių požiūrių taip vadinimas sutarties performinimas realizuojamas gan paprastai. WDA sistemoje įvedėme papildomą poliso statusą (pasiūlymas, registruotas polisas, patvirtintas polisas, pakeistas polisas), tokiu būdu kai vartotojas atidaro sudarytą sutartį ir nurodo, kad nori atlikti sutarties performinimo veiksmą, sistema leidžia jam koreguoti visus laukus (pagal jo kompetencijos ribas) ir saugojimo metu prie pradinės sutarties nurodo pakeitimo statusą, o prie naujai sudarytos nurodo pirminės sutarties numerį, t.y. keičiamos sutarties numerį. Tokiu būdu gaunamas ryšys tarp visų konkrečios sutarties variantų, t.y. turime pilną sutarties istoriją.

4.3. Transakcijų valdymo mechanizmas

Transakcijų valdymas užtikrinamas būtent XML pagalba, t.y. interfeiso dalyje visa įvesta informacija renkama į XML dokumentą, todėl nereikia kas kart atlikti rašymo veiksmą į DB. XML struktūra suteikia galimybę išsaugoti polisą ir tvarkingai jį įdėti į DIS duomenų bazės lentelės vienos transakcijos metu. Šiam tikslui yra naudojama tarpinė DB lentelė, kuri saugo įvestas sutartis XML formatu. Ši lentelė yra užpildoma tik skaičiavimo ir poliso registravimo metu. Kelis kartus skaičiuojant įmoką, naujas įrašas DB neatsiranda, tiesiog vykdomas einamojo įrašo atnaujinimas, o registravimo veiksmas vienam polisui gali būti atliktas tik vieną kartą. Išimtis yra klaidų pranešimų atvejais, tam, kad kontroliuoti vartotojų veiksmus, bei turėti galimybę kompetentingai atsakyti į užduotus klausimus, kodėl sistema jiems neleidžia išsaugoti poliso, kiekvieną kartą rašomas naujas įrašas, t.y. saugomas XML dokumentas, kuris iššaukė klaidos pranešimą. Tokiu būdu mes lengvai galime surasti konkretaus vartotojo padarytas klaidas, bei paaiškinti, kur jis daro klaidas.

Šiam tikslui, būtent klaidų auditavimui ir vartotojų administravimui buvo sukurtas papildomas įrankis, kuris leidžia „Pagalbos skyriui“ lengvai rasti vartotojus bei jų polisus ir padėti jiems susidoroti su iškilusiomis problemomis.

Tik užregistravus polisą, duomenys yra replikuojami į pagrindinės sistemos DB lenteles. Šiuo atveju duomenys imami būtent iš išsaugoto XML dokumento, o ne iš vartotojo įvedimo formos, t.y. visas veiksmas atliekamas DB pusėje kas yra žymiai efektyviau ir saugiau. Net gi nutrūkęs

ryšys šiuo atveju nėra problema, kadangi jeigu buvo išsaugotas XML dokumentas visos paskesnės operacijos vykdomos duomenų bazės pusėje, nepriklausomai nuo to ar klientas yra on-line ar off-line režime.

4.4. Delninio kompiuterio interfeisinis priedas

Mes pagalvojame, kad tokia sistema galima naudoti ne tik personaliniuose kompiuteriuose, bet ir naudojant mobiliuosius telefonus, kadangi pasirinktas paprastas techninis sprendimas palaikomas šiuolaikiniuose telefonuose. Tie patys principai užtikrina tokio praplėtimo veikimą. Reikia tik pakoreguoti interfeisinę dalį, kadangi ne visos mobiliosios operacinės sistemos palaiko kūnus (*ifraime'us*) bei reikėtų tiesiog supaprastinti įvedimo formas tam, kad patogiau būtų dirbti ant mažo ekrano. Šis modulis vis dar vystomas, bet iš esmės preliminarūs testavimai jau buvo atlikti ir jokių neišsprendžiamų problemų nebuvo aptikta.

4.5. Integracija su išoriniais tiekėjais

Šiuo metu WDA sistema palaiko ryšius su keliomis išorinėmis sistemomis, kurie teikia duomenis apie draudėją arba draudimo objektą. Visas duomenų apsikeitimas vykdomas XML ir servletų pagalba. Taigi WDA pasiima duomenis iš „Lietuvos banko“ – pirmasis vartotojas padaręs užklausą apie šiuos dienos valiutos kursą, paleidžia procesą, kuris grąžina Lietuvos banko duomenis einamajam laiko momentui, bei užpildo Valiutos lentelę tai dienai. Tokiu būdu kiekvienas prisijungęs vartotojas neturi atlikti šio veiksmo. Taip pat WDA turi sąryšį su Lietuvos Motoriniu Biuru, kas yra būtina kiekvienai draudimo bendrovei, t.y. pagal įstatymą draudimo bendrovės privalo teikti duomenis biurui, bei turi ryšį su „Regitra“ iš kurios pasiima duomenis apie draudžiamas transporto priemones. Dabar autoriai dirba ties integracija su „Lietuvos Respublikos gyventojų registro centru“ bei „Kadastro“ duomenų bazėmis. Iš pirmojo šaltinio gausime asmens duomenis, o pagal antrąjį galėsime identifikuoti asmens gyvenamąją vietą, kas smarkiai pagreitins draudžiamų asmenų duomenų įvedimo procesą.

Su visais išoriniais duomenų tiekėjais yra suderinti skirtingi duomenų apsikeitimo protokolai, kadangi teikiama informacija yra saugoma skirtinguose formatuose, o mūsų tikslas ją unifikuoti, t.y. sukurti tam tikrus bendrus reikalavimus visiems duomenims, kurie pasiekia draudiko duomenų bazę. Todėl nėra kažkokio bendro mechanizmo kuris tiktų visoms išorinėms kompanijoms, būtent dėl šios priežasties kiekvienas naujo šaltinio prijungimas yra svarstomas atskirai, bei jam keliami tam tikri reikalavimai. Pagrindinė tendencija, bei noras yra kiek įmanoma daugiau informacijos apsikeitimą vykdyti per tam tikrus sukurtus web-servisus, bet ne visi išoriniai šaltiniai turi tokias galimybes. Tokiais atvejais duomenų apsikeitimas vykdomas kitais būdais, tarkim Oracle vidinės procedūros iškviatimo būdu.

4.6. Poliso išdavimo procesas

Kadangi draudimo bendrovės produktas yra draudimo apsaugos pardavimas, t.y. draudimo poliso išdavimas klientui, šis procesas yra vienas svarbiausių kiekvienai draudimo bendrovei. Todėl tai yra savotiškas kiekvienos bendrovės patyrimas. Šiuolaikinės technologijos leidžia smarkiai pagreitinti pardavimo procesą, t.y. supaprastėjo dialogas tarp kliento ir draudiko, sutarties išdavimas (spausdinimas) vykdomas iš sistemos, o ne išrašomas ranka, arba iš viso eliminuojamas agentas iš sutarties pasirašymo ir išdavimo proceso, kitaip tariant supaprastėja pats poliso išdavimo procesas. Šiuo metu pagrindinė bet kurios draudimo bendrovės užduotis yra pereiti nuo ranka rašomų polisų, prie polisų išrašytų per tam tikrą draudimo sistemą. Tik tai pilnai automatizavus šį procesą draudimo bendrovė galės pilnai valdyti prisiimtas rizikas, bei teisingai elgsis su klientais, t.y. bus kontroliuojamos įmokos, kadangi tik tai tokiu atveju įmoka bus apskaičiuota pagal tarifavimo sistemą, o ne žmogus – agentas savo subjektyvia nuomone nustatys poliso kainą.

Vystant WDA mes stengėmės patobulinti egzistuojantį draudimo procesą ir bent jau iš techninės pusės sumažinti poliso išrašymo laiką iki minimumo (mūsų tikslas TPVCA poliso išdavimo laikas neturi viršyti 3 min.). Dabar pabandydysime detaliau aprašyti mūsų naudojamą procesą.

Kadangi WDA kuriama kaip sistema, kuria gali naudotis pardavėjai arba patys draudėjai, todėl mes atskirsime autorizavimo modulį nuo tiesioginio pardavimo proceso.

- Vartotojas prisijungia prie sistemos, jis mato draudimo produktus, kuriuos turi teisę pardavinėti arba turi teisę apsidrausti (savitaros portalo atveju);
- Vartotojas pasirenka norimą draudimo produktą;
- Užpildo draudėjo, bei poliso duomenis, t.y. bendra visiems produktams dalis. Draudėjo įvedimas – tai asmens kuris sudarinėja sutartį duomenys, kuriuos sistema pakrauna iš duomenų bazės arba išorinių registrų. Tokiu būdu atvejų kada pačiam vartotojui reikia suvesti šiuos duomenys ranka skaičius yra labai mažas (tik asmenys kurie neturi galiojančio asmens tapatybės dokumento arba užsieniečiai, kurie dar nesidraudė bendrovėje). Poliso duomenys užpildomi automatiškai, bet juos galima koreguoti priklausomai nuo draudėjo ar apdraustųjų poreikių. Kadangi ši dalis yra bendra visiems draudimo produktams, vartotojams tereikia įsiminti vieną procesą, o ne keliasdešimt, kiekvienam draudimo produktui savo;
- Jeigu draudėjo bei poliso duomenys buvo sėkmingai įvesti, sistema leidžia pereiti prie draudimo objekto (produkto) įvedimo, kitų atvejų informuoja vartotoją apie nustatytas klaidas;

- Vartotojas suveda produkto duomenis, pasirenka norimas draudimines rizikas ir riziką įtakojančius faktorius bei paskaičiuoja įmoką. Jeigu nebuvo aptiktos vartotojo įvedimo klaidos, sistema gražina poliso įmoką, kitu atveju gražinamas klaidos pranešimas;
- Vartotojas registruoja polisą, jeigu draudėjui tinka pasiūlyta draudimo įmoka. Registravimo metu vartotojui išvedama poliso duomenų suvestinė, t.y. prieš registruojant galima dar kartą patikrinti ar nebuvo padarytos kokios nors įvedimo klaidos (tokiu būdu bandoma sumažinti žmogiškojo faktoriaus klaidas). Jeigu vartotojas patvirtina duomenų korektiškumą vykdomas sutarties validavimas (verslo reikalavimų tikrinimas), jeigu reikia autorizavimas bei registravimas.
- Sėkmės atveju vartotojui gražinamas pranešimas apie sėkmingą poliso registravimą ir jis pereina prie sutarties spausdinimo proceso.

Dauguma sistemų dokumentų spausdinimo procesą atskiria nuo poliso įvedimo proceso. Autorių manymų tai nėra labai korektiška, kadangi tik tai tuo atveju, kai žmogus pasirašo sutartį ir susipažįsta su visomis taisyklėmis polisas skaitosi įsigaliojęs. Todėl WDA sistemoje polisų spausdinimo procesas yra neatsiejama poliso išdavimo proceso dalis. Tokiu būdu mes užtikrinam kliento saugumą, t.y. tik po to kai vartotojas atspausdina visus būtinus dokumentus sutartis yra patvirtinama. Tam, kad pagreitinti vartotojų darbą, bei sumažinti bendrovės aptarnaujančio personalo darbą, WDA sistemoje realizuotas automatinis blankų gadinimo procesas, taip nesėkmingai atspausdinus dokumentą, pats vartotojas gali jį sugadinti ir iškart panaudoti kitą. Supaprastintas spausdinimo procesas atrodo šitaip:

- Vartotojas atidaro blanko nurašymo formą;
- Sistema automatiškai nustato koks blankas dabar bus spausdinamas ir praneša vartotojui kokį blanką reikia įdėti į spausdintuvą;
- Vartotojas nurodo blanko seriją ir numerį (jeigu blankas yra griežtos atskaitos) ir jį nurašo. Jei blanko nurašyti nepavyko sistema gražina klaidos pranešimą;
- Suformuojamas spausdinamas dokumentas;
- Vartotojas inicijuoja patį spausdinimą ir atsako į sistemos užduotus klausimus apie proceso eigą. Yra galimi trys variantai:
 - Blankas sėkmingai panaudotas,
 - Blankas nepanaudotas,
 - Blankas sugadintas;
- Priklausomai nuo rezultato sistema pasiūlo:
 - Atspausdinti sekantį dokumentą arba jei tokių nėra patvirtina sutartį;
 - Pakartoti dokumento spausdinimą ant to pačio blanko;
 - Pakartoti dokumento spausdinimą ant naujo blanko.

Išsamius poliso išdavimo bei spausdinimo procesus galima surasti priede Nr. 2.

4.7. API ir DB ypatybės

Sukurtoje WDA sistemoje pagrindinę vietą užima sistemos API. Sistemos API padeda konkrečiau produktui kūrėjui abstrahuotis nuo sistemos veikimo ir susitelkti visą dėmesį į funkcionalumo įgyvendinimą. Taip pat API siūlo naudoti standartines pagalbines funkcijas.

Sistemos API susideda iš trijų pagrindinių paketų: poliso, draudėjo ir pagalbinių funkcijų.

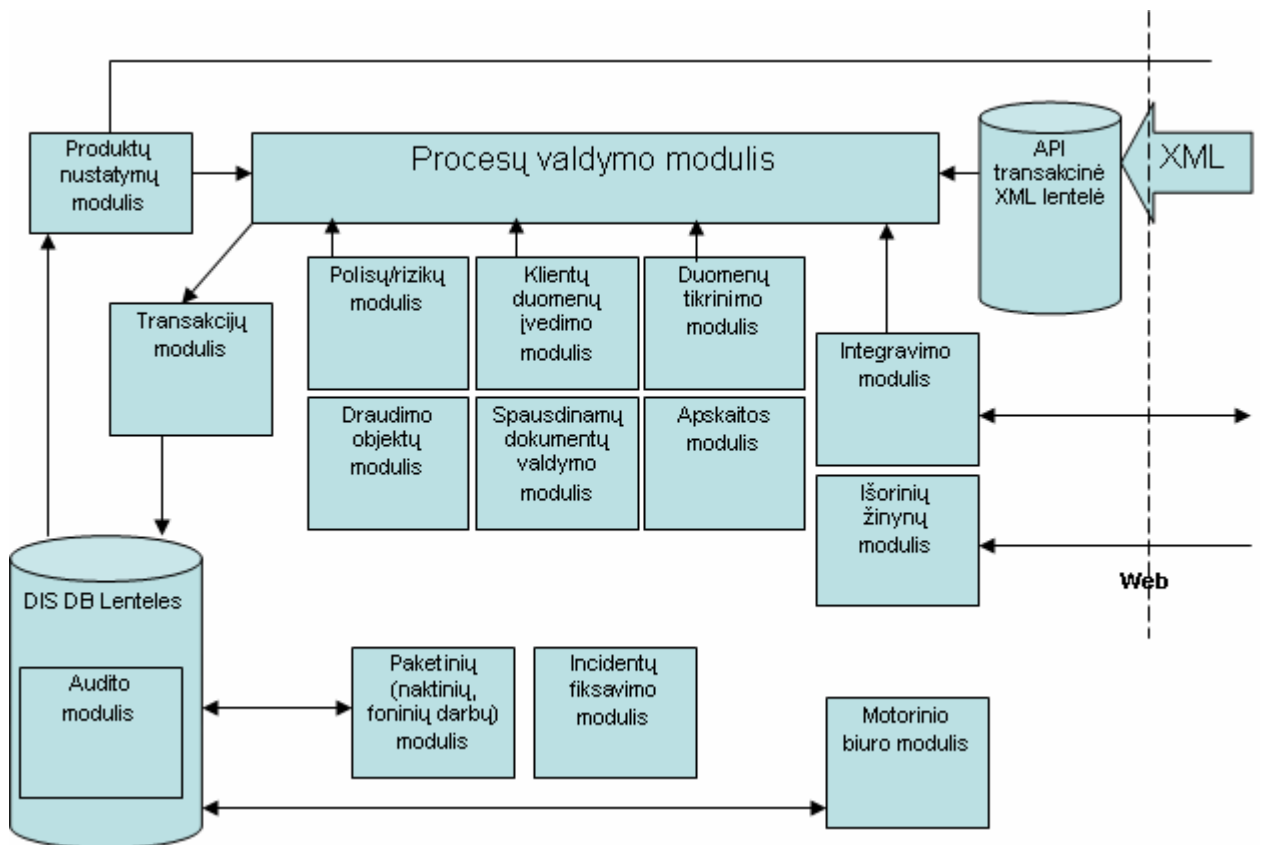
Pagalbinių funkcijų naudojimas padeda standartiškai prisijungti prie sistemos, paleisti sistemos tvarkymo (DEBUG) ir kontrolės (LOG) modulius, nustatyti kiekvienam vartotojui atitinkamas teises. Kita paketo dalis yra laukų tikrinimo ir įmokų/įmokų grafiko skaičiavimas. Realizuojant konkretų produktą užtenka tik kreiptis į pagalbines funkcijas ir galima negalvoti apie įmokų apvalinimą, paskirstymo tarp rizikų ar mokėjimo grafiką. Aišku, kad pagalbinių funkcijų pakete taip pat yra keletas funkcijų kurios tiesiog padeda patogiau programuoti.

Draudėjo pakete realizuotos trijų rūšių funkcijos: informacijos apie draudėją gavimas, atnaujinimas ir tikrinimas. „Gavimo“ funkcijos naudojamos ypač dažnai, nes negalima įvesti nei vienos draudimo sutarties be draudėjo informacijos. Su šios rūšies funkcijomis galima lengvai ir greitai gauti informaciją apie draudėjo kodą, vardą/pavardę bei adresus nepriklausomai nuo to ar ieškomas draudėjas fizinis ar juridinis asmuo.

Kita rūšis, tai duomenų atnaujinimo funkcijos. Jos kviečiamos kartu su gavimo funkcijomis. Žiūrint iš konkrečiau produkto realizavimo pusės, šias funkcijas reikia kviešti visada kai norima išsaugoti informaciją apie draudėją, nesvarbu ar toks draudėjas jau yra bazėje ar įvedamas pirmą kartą.

Trečia rūšis iš eilės, bet pirma pagal svarbumą tai tikrinimo rūšis. Kitos rūšys tik palengvina „gyvenimą“, o ši yra būtina. Tikrinimo funkcijos yra atsakingos už įvedamų duomenų korektiškumą ir vientisumą, kas leidžia turėti „teisingą“ informaciją apie draudėjus. Neturint teisingos informacijos kompanija negalės pasiūlyti papildomų paslaugų, tokių kaip automatinis polisų, priminimų ir skolų pranešimų siuntimas paštu, pasiūlymų teikimas atsižvelgiant į kliento draudimo portfelį.

Paskutinis API paketas – polisas. Šio paketo funkcijos atsakingos už techninį poliso įrašymą į pagrindinę duomenų bazę. Įrašant vykdoma duomenų korektiškumo kontrolė, versijų keitimas, transakcijų tikrinimas. Jeigu sistemos vartotojas tvirtina sudarytą polisą turi įvykti globalus duomenų patvirtinimas (commit) kurio metu įrašomi duomenys apie apdraustą produktą, rizikas ir draudimo objektą.



Pav. 6 „WDA API architektūra“

4.8. Atviras interfeisas į verslo sistemą (pernešamumas)

WDA sistemos architektūra remiasi XML dokumentų naudojimu. Kaip jau minėjome anksčiau, toks XML dokumentų naudojimas leidžia užtikrinti kokybišką transakcijų mechanizmo darbą, bei kas svarbiausia suteikia sukurtai sistemai pernešamumo savybę. Bet koks įvestas polisas yra išsaugomas tarpinėje XML lentelėje ir tik poliso registravimo metu, visi poliso duomenys yra replikuojami į pagrindinę sistemą arba jeigu tokios nėra į WDA sistemos lenteles. XML dokumento struktūra susideda iš kelių pagrindinių dalių:

- Draudėjas;
- Apdraustasis;
- Poliso duomenys;
- Draudžiamo objekto duomenys;
- Draudžiamos rizikos;
- Mokėjimų duomenys.

Kiekvienas XML dokumento žymės pavadinimas atitinka duomenų bazės lentelės stulpelį, t.y. nurodytas mazgas yra lygus lentelės stulpelio pavadinimui, o nurodyta žymės reikšmė yra lygi būtent tai reikšmei, kuri turi būti įrašyta į duomenų bazę. Stulpelio reikšmė yra nurodoma su jai reikalinga konvertavimo funkcija, pvz.

```
<sutartis>  
  <polisas>  
    <stulpelisA>  
      to_char('reiksme1')  
    </stulpelisA>.  
    <stulpelisB>  
      reiksme2  
    </stulpelisB>.  
  </polisas>  
</sutartis>
```

Tokių būdų WDA gali būti lengvai adaptuojama prie bet kurios kitos draudimo informacinės sistemos, tiesiog atitinkamai pakeičiant XML žymių reikšmės. Priede Nr. 3 yra pateiktas naudojamo XML dokumento pavyzdys su paaiškinimais.

5. Realizavimo ypatybės

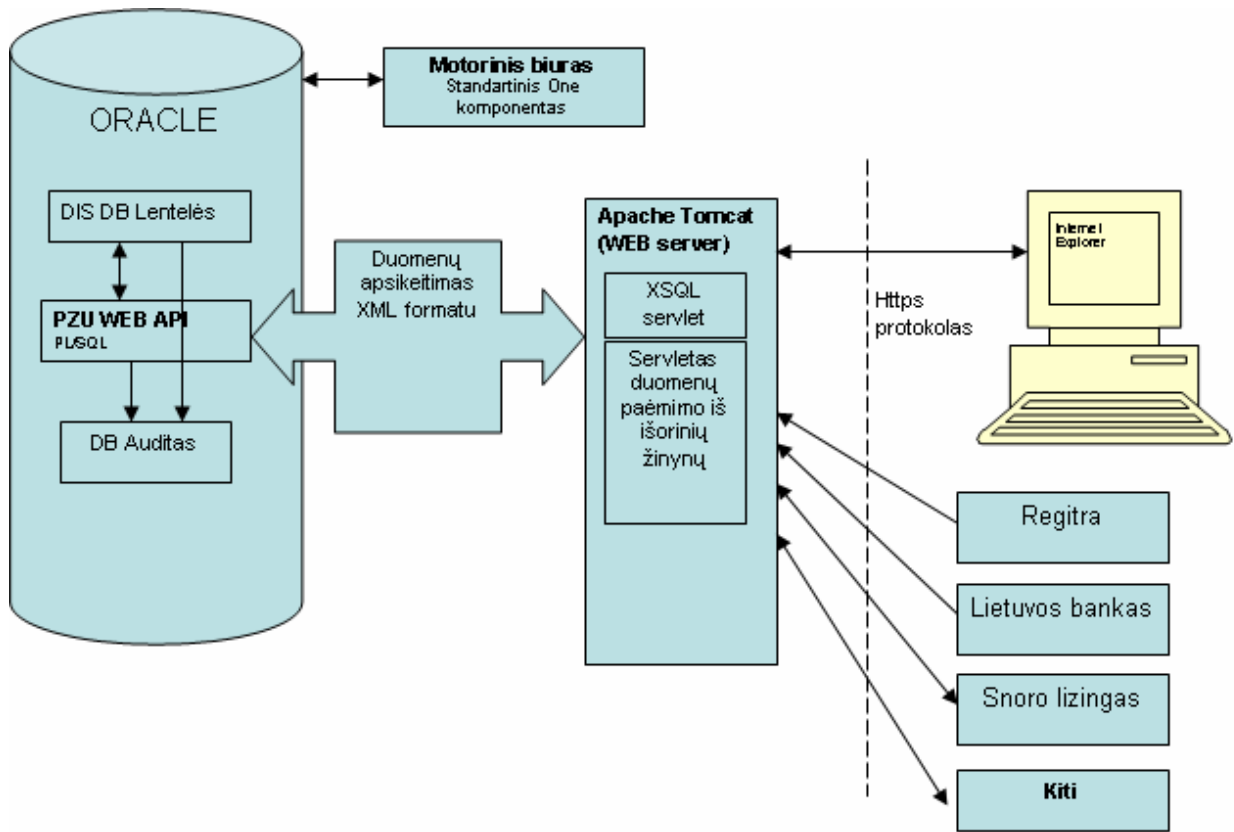
Šioje dalyje pabandysime atkreipti skaitytojo dėmesį į įdomiausias bei kebliausias sistemos kūrimo vietas, paaiškinsime kaip ir kodėl buvo priimti tam tikri sprendimai, bei identifikuosime savo padarytas klaidas ir nurodysime jų sprendimo būdus.

Naudojamas RAD įrankis buvo sukurtas naudojant paprastą HTML interpretavimo mechanizmą. Bet kokia vartotojo nupaišyta forma, buvo konvertuojama į HTML priimtina formą su visais pasirinktais stiliais, pozicijomis, duomenų tikrinimo funkcijomis ir t.t.. Bet toks sprendimas yra pakankamai griezdiškas, bei nėra labai lankstus, kadangi kiekvieno naujo elemento įdiegimas, t.y. elemento kuris nebuvo numatytas interpretatoriuje iš pat pradžių reikalauja pakankamai daug programavimo darbų. Todėl natūralus noras buvo surasti išeitį, kuri leistų sumažinti pačius programavimo darbus, jau sukurtų formų pakrovimo laiką, bei jų išeities kodo dydį. Atlikti tyrimai parodė, kad XML duomenų salos (XML data island) siūlo mechanizmą, kuris leidžia surišti XML dokumentus su HTML formomis, bei atvaizduoja formų duomenis. Todėl RAD formų dizaineris sukonstruotas būtent XML data island naudojimo principu, kuris leidžia kurti formas su žymiai mažesniais išeities tekstais, negu paprastas HTML puslapis. Dizainerio pagalba sukurtos formos yra pastebimai mažesnio dydžio, todėl žymiai greičiau pasikrauna kas yra labai svarbu galutiniam vartotojui. Be to mažesnė klientinė aplikacijos dalis sumažina ir klaidų tikimybę, kadangi pirmiausia nereikia atlikti programavimo darbų, t.y. eliminuojamas žmogiškasis faktorius. Taip pat mažesnis puslapio dydis užtikrina mažesnę siunčiamo paketo dydį, ko pasekoje ženkliai sumažėja prarastų paketų tikimybė, kas padidina sistemos veikimo stabilumą.

Antrasis ne mažiau svarbus ir įdomus WDA sistemos techninis sprendimas yra XML dokumentų naudojimas. Jau ne kartą šnekėjome apie visus privalumus, kuriuos suteikia toks sprendimas, bet tai yra vienas iš pagrindinių projektavimo dalies nuopelnų, kuris padėjo sumažinti sistemos apkrovimą, kontroliuoti transakcijų mechanizmą, abstrahuotis nuo pagrindinės sistemos, t.y. leido sukurti API, kurio pagalba WDA sistema įgavo pernešamumo savybes. Visas sunkiausias darbas buvo tvarkingai aprašyti XML dokumento struktūrą, bei duomenų rašymą į duomenų bazės lenteles pagal turimą XML dokumentą. Šiam tikslui buvo aprašyta Oracle vidinė procedūra, kuri pagal nurodytą kelią ištraukia iš išsaugoto dokumento duomenis ir pagal žymės pavadinimą automatiškai pakrauna įrašą į jam skirtą DB vietą. Toks sprendimas supaprastina duomenų apsikeitimo schemą, leidžia labai lengvai atlikti bet kokius pakeitimus, t.y. naujo elemento įdėjimą arba seno ištrynimą. Kliento dalyje paprasčiausios JS sukurtos bibliotekos pagalba formuojamas XML dokumentas pagal vartotojo įvestus duomenis, kuris yra perduodamas į DB XSQL tarnybinio įskiepio pagalba. Taip pat ši biblioteka leidžia pakrauti XML duomenis į nurodytą HTML formą. Jau sistemos realizavimo metu susidūrėme su

problema, kadangi XSQL įskiepis apdoroja visus kintamuosius tik eilutės formatu, bei apriboja perduodamos eilutės ilgį iki 4000 simbolių. Tokie apribojimai nėra labai aktualūs, kadangi dauguma įvedamų poliso duomenų yra mažesni. Bet tam, kad padaryti sistemą, kuri nepriklausytų nuo tokių niuansų, bei stabiliai veiktų su bet kokiais duomenimis ir bet kokiais duomenų kiekiais, buvo sukurtas mechanizmas kuris iš turimų duomenų suformuoja tokius blokus, kurie atitinka XSQL keliamus reikalavimus ir yra perduodami ar priimami pagal eilės tvarką ir tik tai tuo atveju kai visi duomenis sėkmingai buvo apdoroti XML dokumentas yra išsaugomas į lentelę.

Trečiasis aspektas vertas dėmesio yra API struktūra. API yra suprojektuotas bei užprogramuotas tokiu būdu, kad 90% visų atliekamų operacijų yra pilnai standartizuotos, net gi kiekvieno produkto įmokos nustatymo procesas yra gan automatizuotas. Kaip jau buvo aprašyta aukščiau, API susideda iš standartinių, pagalbinių ir produkto specifinių funkcijų ir paketų. T.y. bendra API dalis keičiama tik tai tuo atveju kai daromi esminiai sistemos funkcionalumo pakeitimai, produkto keitimo atvejais keičiamas konkretaus produkto realizacija. Produkto dalis susideda tik iš dvejų dalykų: įmokos skaičiavimo algoritmo realizavimas ir objekto duomenų tikrinimo procedūros. Visi kiti veiksmai yra atliekami pagal bendras aprašytas standartiniuose vidiniuose paketuose procedūras. Skaičiavimo algoritmo iškvietimas yra pilnai automatizuotas, t.y. pagal tam tikrą produkto kodą, sistema iškviečia būtent nurodyto produkto skaičiavimo algoritmą, t.y. vartotojas iškviečia vieną įmokos skaičiavimo procedūrą, kuri pati nukreipia duomenis į jiems skirtus apdorojimo algoritmus.



Pav. 7 „WDA sistemos architektūra“

6. Sistemos įvertinimas

Kadangi vienas iš pagrindinių šio tiriamojo darbo tikslų yra sukurti patikimą, stabiliai ir greitai veikiančią Web draudimo sistemą, autoriai turėjo įvertinti ar siūlomas technologinis sprendimas, ar patobulinti veiklos procesai atitinka keliamus reikalavimus ar ne. Šioje dalyje bus pateiktas išsamus sistemos įvertinimas, palyginimas su jau aprašytais sistemomis, nurodytos stipriausios bei silpniausios WDA vietos.

Visų egzistuojančių sistemų pasirinkti technologiniai sprendimai jau buvo aukščiau aprašyti. Kokie iš jų yra geresni ar blogesni yra skonio reikalas, todėl šį klausimą ir pasirinkimą mes paliekam skaitytojui. Svarbiausia, kad pasirinkti technologiniai sprendimai palaikytų sistemos veikimo greitį, bei teisingai operuotų su turimais duomenimis. Todėl mes atliksime funkcinę egzistuojančių sistemų analizę ir patikrinsime ar pasisekė autoriams įgyvendinti savo tikslą.

Iš pradžių palyginsime WDA su pagrindinėmis egzistuojančiomis draudimo verslo sistemomis pagal aukščiau aprašytus draudimo sistemai keliamus reikalavimus, t.y. praplėsime atliktą analizę įtraukdami į ją WDA sistemą.

Funkcionalumas	TIA	DIS	WDA
<i>Autorizacijos modulis</i>	+	-	+
<i>Validavimo modulis</i>	+	+	+
<i>Kvotų posistemė</i>	-	-	+
<i>Tarifikacijos modulis</i>	+/-	+/-	+
<i>Sutarties administravimo galimybės</i>	+	-	+
<i>Auditas</i>	+/-	+/-	+
<i>Produktų dizaineris</i>	-	+	+
<i>Apskaitos modulis</i>	+	-	+
<i>Klientų segmentavimo galimybės</i>	+	-	+
<i>Vieningas draudžiamų objektų/asmenų registras</i>	-	-	+/-
<i>Automatinis replikavimas į brokerių sistemas</i>	-	-	+/-
<i>Naktiniai procesai</i>	+	+	+
<i>Profailų modulis (ribotas teisių suteikimas)</i>	+/-	-	+
<i>Automatinis sutarčių pratęsimas</i>	+	-	+/-
<i>Sudėtinių produktų pardavimas</i>	+/-	+/-	+
<i>Ryšys su išoriniais duomenų tiekėjais</i>	+	+/-	+

Lentelė 3 „3-jų sistemų funkcionalumo palyginimas“

Iš pateiktos funkcionalumo palyginimo suvestinės akivaizdžiai matosi, kad WDA sistema atlieka daugiau verslui reikalingų funkcijų, negu konkurencingos sistemos. Šis faktas yra lengvai paaiškinamas tuo, kad autorių atliktas išsamus draudimo sistemos aprašymo darbas buvo atliktas ne veltui ir tai yra svarbi sistemos projektavimo dalis. Būtent todėl standartinis WDA sistemos funkcionalumas buvo kuriamas atsižvelgiant į visus nustatytus verslo keliamus reikalavimus ir poreikius, bei egzistuojančių sistemų trūkumus.

Kaip matosi iš atliktos analizės WDA sistema palaiko kvotų suteikimo bei administravimo procesus, kas leidžia pagreitinti pardavimo procesą, bei palengvina rizikos vertinimą, kadangi suteikti nuolaidų krepšeliai yra ribojami bei audituojami individualaus vartotojo arba skyriaus lygyje. Įdomesnė vieta yra automatinio replikavimo į išorines sistemas realizavimas, tai yra mechanizmas, kuris leidžia išoriniams vartotojams įrašyti gautus duomenis į savo sistemas. Iš techninės pusės realizavimas yra labai paprastas, kadangi visi duomenys apie įvestus polisus WDA sistemoje saugomi XML formatu su bet kuriuo išoriniu vartotoju, t.y. brokerių, bankų ar lizingo kompanija, WDA sistema per web-servisą apsieičia XML dokumentais, kurie yra lengvai interpretuojami į kliento pageidaujamą formatą. Šioje vietoje pažymėjimas „+/-“ parašytas tik dėl to, kad kol kas sistema turi tokį apsieikimo protokolą tik su vienu vartotoju ir dar nėra pilnai standartizuota. Kas liečia automatinio sutarčių pratęsimo procesą – šis procesas yra pilnai įdiegtas į WDA sistemos API logiką, bet dėl turimų duomenų nekorektiškumo, turint omenyje adresų struktūras šis funkcionalumas kol kas nėra naudojamas.

Ne mažiau svarbus yra sistemos našumo įvertinimas, t.y. turi būti atilikti testavimai, kurie parodys kiek vartotojų vienu metu galės dirbti su sistema, patikrinti servisinę sistemos pusę, parinkti tinkamą blokų sudarymo (cluster) strategiją. Šiam tikslui autoriai palygino WDA sistemą su TIA sistemos WEB praplėtimu (toliau vadinsime TIA WEB).

	TIA WEB	WDA
<i>Vartotojų skaičius</i>	3000	2600 +
<i>Vienalaikių prisijungimų skaičius</i>	~ 500	~ 400
<i>Blokinis mechanizmas įjungiamas nuo</i>	100	450
<i>TPVCA poliso įvedimo laikas</i>	4,5 min.	2 min.
<i>TPVCA poliso įvedimo operacijų skaičius</i>	~ 20	~ 20
<i>TPVCA polisui užklausių į DB skaičius</i>	~ 15	8
<i>Įvedimo klaidų tikimybė</i>	0,032	0,02
<i>Maksimalus leistinas apkrovimas</i>	900	1300

Lentelė 4 „Web sistemų našumo palyginimas“

Kaip matosi iš atliktos analizės dabartinis abejų sistemų vartotojų, bei vienalaikių prisijungimų skaičius yra maždaug vienodas. O turimas skirtumas yra lengvai paaiškinamas keliais faktais. Pirmiausia, WDA sistema buvo paleista tik prieš pusę metų, kai TIA WEB sistema yra naudojama rinkoje jau virš 3 metų. Antrasis argumentas yra sistemas naudojančių įmonių dydžiai. Blokinis mechanizmas, t.y. prisijungusių vartotojų dalijimas į blokus pagal darbo stotis, WDA sistemoje prasideda tik nuo 450, t.y. dabartiniu metu sistema dirba tik su vienu web serveriu, tuo tarpu TIA WEB sistema kiekvienam 100 prisijungimų sukuria naują serverį ir šiuo metu jų yra 6. Eksperimentiniu būdu buvo apskaičiuotas TPVCA poliso įvedimo laikas, kur akivaizdus lyderis yra WDA sistema. Vienas pagrindinių WDA privalumų yra sumažintas užklausų į duomenų bazę skaičius. Kas leidžia smarkiai pagreitinti įvedimo laiką, bei sumažinti duomenų bazės apkrovimą. Lyginant įvedimo klaidų tikimybes buvo įvertinamas sistemų interfeisų patogumas ir aiškumas, buvo įvertinama kiek vartotojų iš viso dirbančių su sistema linkę daryti klaidas, kiek klaidų vieno poliso įvedimo metų daro vartotojai bei koks santykis visų įvestų polisų su polisų skaičiumi kuris buvo įvestas su klaidomis.

Išvados

Atliktas tiriamasis darbas rodo, kad šiuolaikinėse draudimo informacinėse sistemose yra daug trūkumų ir jos ne visiškai atitinka draudimo verslo poreikius. Būtent šis faktas įrodo pasirinktos temos aktualumą. Kadangi draudimas yra specifinė veiklos sritis, todėl nėra labai daug literatūros ar būtent konkrečių veiklos informacinių sistemų analizės ir metodikos. Padaryta egzistuojančių sistemų analizė padėjo identifikuoti silpnąsias sistemų vietas, bei privertė susimąstyti apie šių klaidų ar nepatogumų eliminavimą mūsų siūlomame sprendime. Pagrindinis užsibrėžtas šio tiriamojo darbo tikslas buvo sukurti metodologiją pagal kurią galima būtų lengvai suprojektuoti draudimo sistemą, kuri atitiktų visus pagrindinius draudimo bendrovės poreikius. Sukurta WDA sistema parodo, kaip tai galima padaryti, bei siūlo vieną iš galimų techninių sprendimų. Sekantis svarbus aspektas yra Internetinių technologijų naudojimas, t.y. sukurtos sistemos mobilumas. Kaip rodo praktika bei surinkti statistiniai duomenys polisų išdavimo skaičius išaugo, nes šiuo metu vartotojai (agentai ir brokeriai) gali dirbti ir iš namų ar bet kurios kitos jiems patogios vietos. Taip pat galima pabrėžti kad autoriams pavyko įgyvendinti visus tyrimo pradžioje užsibrėžtus tikslus, t.y.

- Buvo atlikta turimų draudimo informacinių sistemų analizė;
- Pavyko išvystyti Web verslo sistemos projektavimo mechanizmą;
- Sukūrėme atviros sąsajos algoritmus;
- Pagerinome poliso įvedimo laiką (TPVCA – 2 min.);
- Apibrėžėme poliso įvedimo procesą;
- Sukūrėme produktų dizainerį, kvotų bei tarifų valdymo modulius;
- Patobulinome techninį poliso išdavimo procesą;
- Atlikome analitinę ir praktinę visų nagrinėjamų sistemų analizę;

Atliktas WDA sistemos palyginimas su jau egzistuojančiais komerciniais produktais parodė, kad indukcinį metodų sukurta metodologija pateisino autorių lūkesčius, kadangi pagal aprašytus sistemos reikalavimus sukurta produktas (savo baziniu funkcionalumu) yra pranašesnis už savo konkurentus. Mokslinis tiriamojo darbo įnašas padės ateityje sėkmingiau ir efektyviau projektuoti informacines draudimo sistemas, nes be naujai sukurtos draudimo aplikacijos, autoriai pasiūlė projektavimo mechanizmą, t.y. aprašė visas būtinas sistemos sudedamąsias dalis, apibrėžė kritinius aspektus, bei optimizavo poliso išdavimo procesą, kuris užtikrina greitą ir patikimą pardavimo bei klientų aptarnavimo procesą.

Šis tiriamasis darbas yra naudingas LR draudimo bendrovėms. Kadangi draudimo verslas yra griežtai reguliuojamas įstatymais, todėl mes savo darbe apsiribojome būtent Lietuvos respublikos draudimo rinka. Šio tiriamojo darbo rezultatas, t.y. sukurta metodologija bei procesais bus naudojamosi sekančiame projekte – savitarnos draudimo portalo kūrimo. Šis modelis jau dabar

yra kuriamas bei įgyvendinamas. Realus produktas pasiūlys draudimo bendrovėms dar vieną būdą teikti kokybiškas paslaugas, dabar ir Interneto erdvėje.

Literatūros sąrašas

- [MTU03] H. Maruyama, K. Tamura, N. Uramoto, +. XML and Java: Developing Web Applications, 2nd Edition. Addison Wesley, 2003.
- [CBB03] P. Clements, F. Bachmann, L. Bass, + . Documenting software architecture. 2nd Edition. Addison Wesley, 2003.
- [CP06] V. Cechanovskij, J. Ptak. Web sistemos funkcionalumas.
URL: <http://homepzu.lt/web/doc/index.htm>. 37 KB, 2006.07.03.
- [Woy03] E. Woychowsky, XML duomenų salos URL: <http://builder.com.com/5100-6371-1058668.html>. 154KB, 2007.03.04
- [WWW01] The Guide to the XML Galaxy. URL: <http://www.zvon.org>. 2006.09.07
- [WWW02] TIA dokumentacija. URL: <http://www.tia.dk>. 2007.01.04
- [WWW03] IPS sistemos. DIS funkcionalumas URL: <http://homepzu.lt/dis/doc/index.htm>. 48 KB, 2006.10.17
- [WWW04] Lietuvos respublikos draudimo įstatymas. 2003 09 18d. Nr. IX-1737.
URL: http://www3.lrs.lt/pls/inter2/dokpaieska.showdoc_l?p_id=281307. 597KB, 2007.02.06
- [WWW05] Lietuvos respublikos draudimo priežiūros komisija, URL: <http://www.dpk.lt/>

PRIEDAI

Priedų sąrašas

Priedas Nr.1	41
Priedas Nr.2	42
Priedas Nr.3	46

Priedas Nr.1**Egzistuojančių sistemų funkcionalumo palyginimas**

Funkcionalumas	TIA	DIS
<i>Autorizacijos modulis</i>	+	-
<i>Validavimo modulis</i>	+	+
<i>Kvotų posistemė</i>	-	-
<i>Tarifikacijos modulis</i>	+/-	+/-
<i>Sutarties administravimo galimybės</i>	+	-
<i>Auditas</i>	+/-	+/-
<i>Produktų dizaineris</i>	-	+
<i>Apskaitos modulis</i>	+	-
<i>Klientų segmentavimo galimybės</i>	+	-
<i>Vieningas draudžiamų objektų/asmenų registras</i>	-	-
<i>Automatinis replikavimas į brokerių sistemas</i>	-	-
<i>Naktiniai procesai</i>	+	+
<i>Profailų modulis (ribotas teisių suteikimas)</i>	+/-	-
<i>Automatinis sutarčių pratęsimas</i>	+	-
<i>Sudėtinių produktų pardavimas</i>	+/-	+/-
<i>Ryšys su išoriniais duomenų tiekėjais</i>	+	+/-

Lentelė 5 „Sistemų funkcionalumo palyginimas“

Sutartiniai ženklai

„+“ - palaiko

„-“ - nepalaiko

„+/-“ – dalinai palaiko






Priedas Nr.2

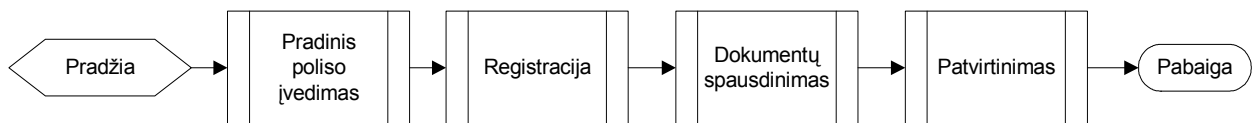
Poliso įvedimo procesas

Darbas su polisu gali būti padalintas į du pagrindinius procesus:

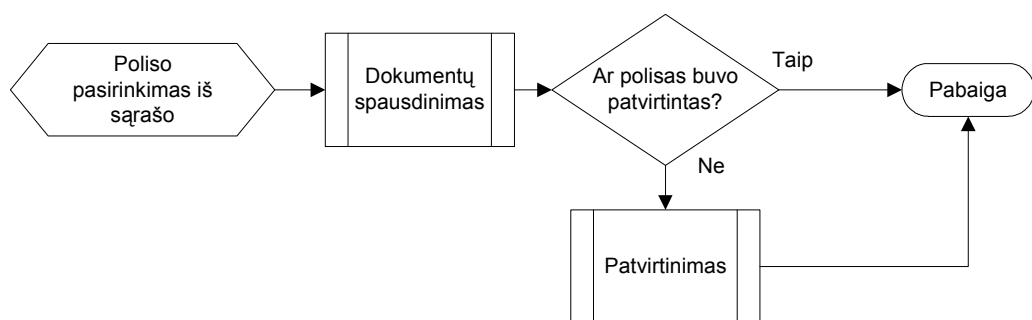
1. Pradinis poliso duomenų įvedimas/dokumentų spausdinimas;
2. Pakartotinas spausdinimas/dublikatų išdavimas.

Sutartiniai ženklai:

	- proceso pradžia;
	- proceso pabaiga;
	- pasirinkimas iš dvejų galimų variantų
	- įterptas procesas
	- duomenų saugojimas

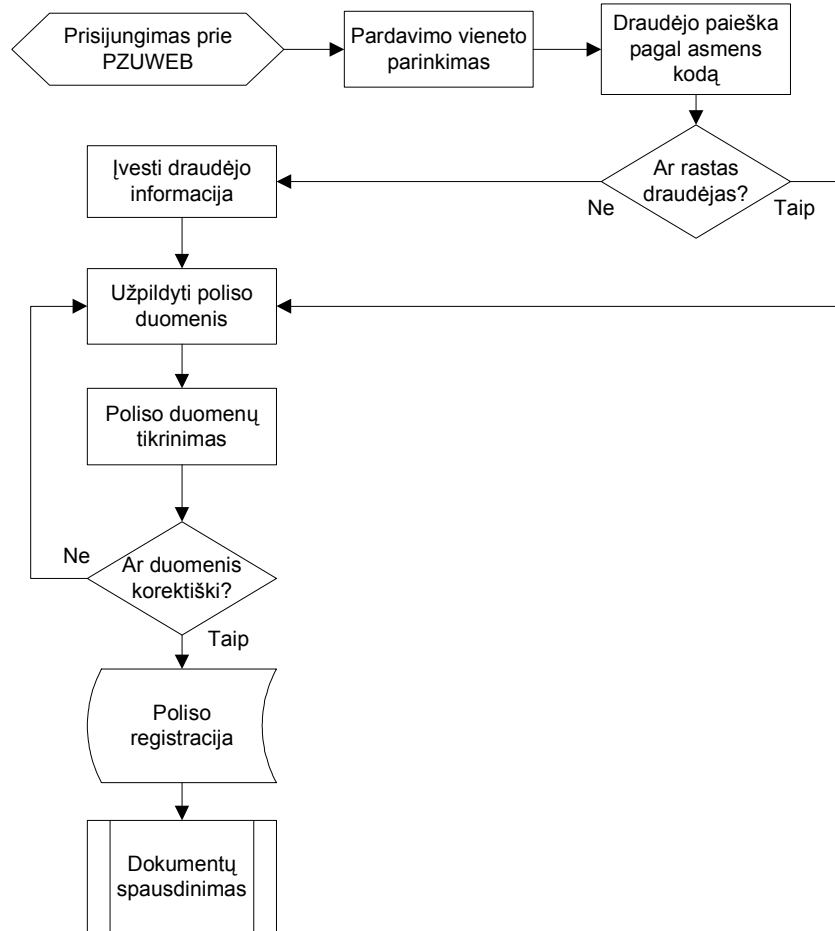


Pav. 8 „Pradinis poliso įvedimas“



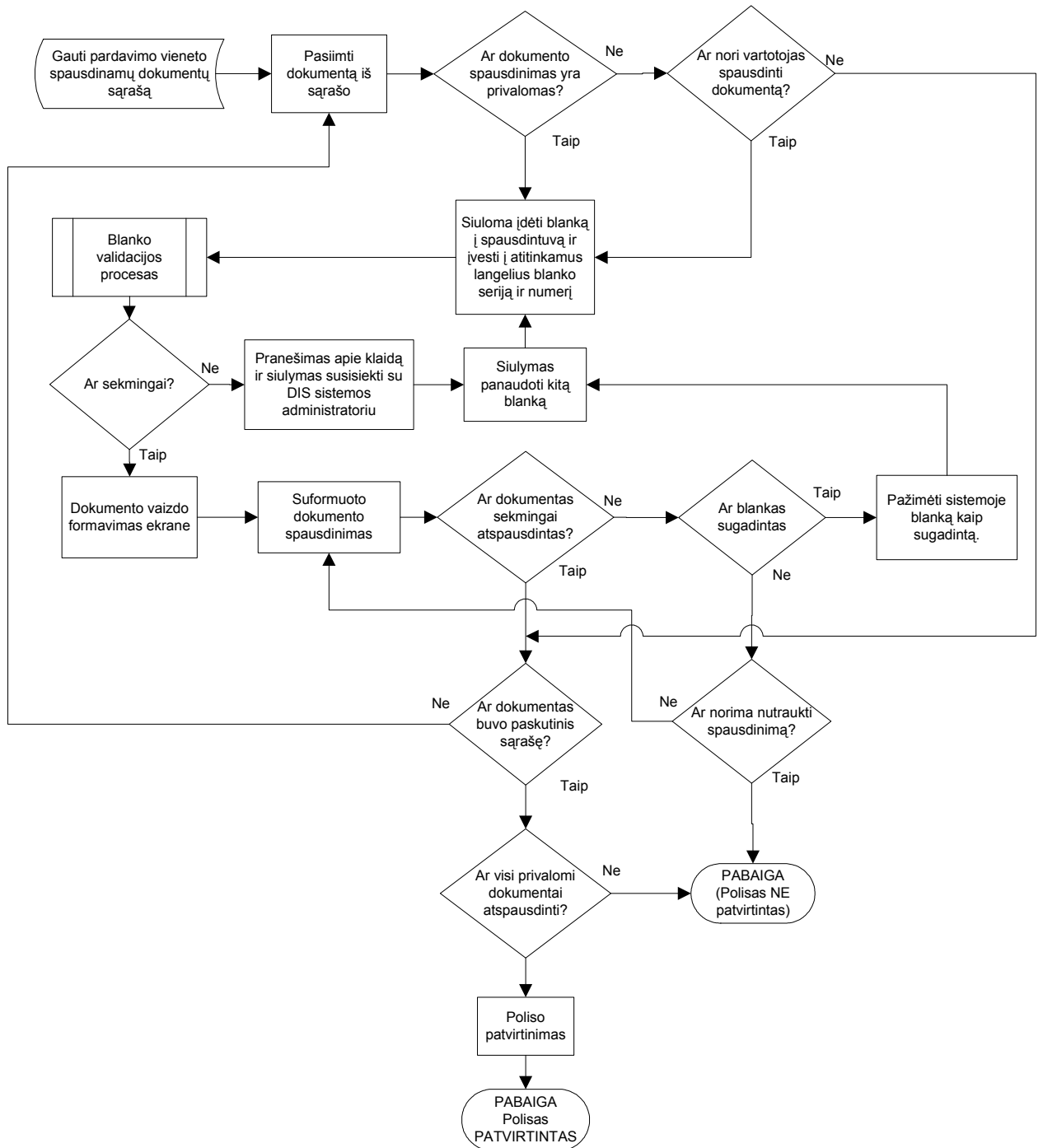
Pav. 9 „Pakartotinas spausdinimas“

Procesas „Pradinis poliso įvedimas“ nurodytas kartu su prisijungimų prie sistemos procesu.



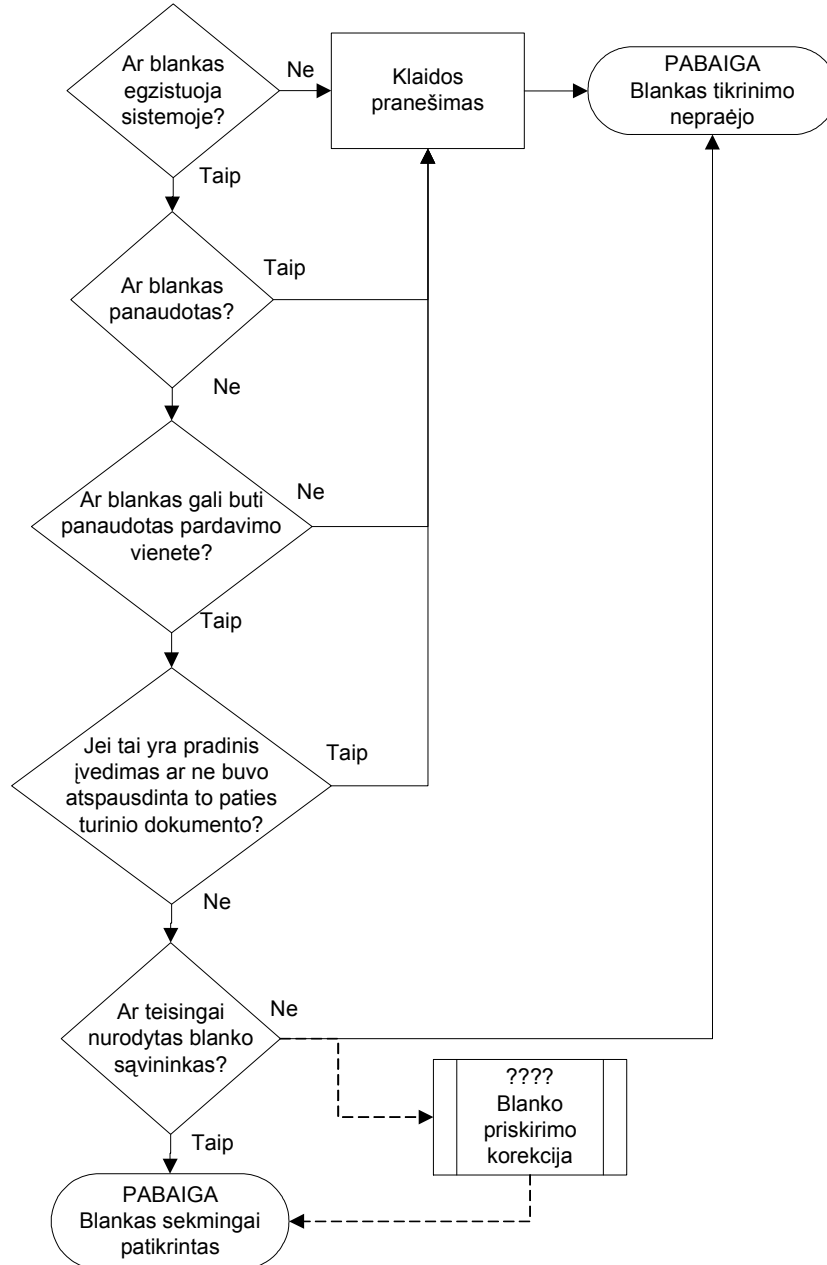
Pav. 10 „Pradinis poliso duomenų įvedimas“

Spausdinimo procesas bus realizuotas kaip vedlys, t.y. dokumentų panaudojimas ar sugadinimas bus atliekamas automatiškai, priimant sprendimą pagal tikrinimus ir vartotojo atsakymus į sistemos klausimus.

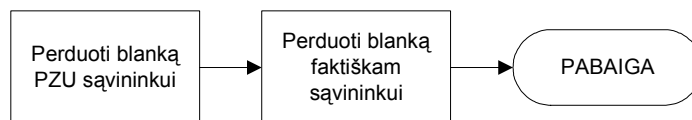


Pav. 11 „Dokumentų spausdinimas/poliso patvirtinimas“

Blanko tikrinimo procesas yra spausdinimo proceso dalys. Piešinyje įterptas siūlomas blanko priskyrimo korekcijos procesas. Sprendimą dėl šio proceso realizavimo priima verslo žmonės.



Pav. 12 „Blanko tikrinimas“



Pav. 13 „Siūlomas automatinis procesas blankų priskyrimo korekcijai“

Priedas Nr.3**Draudimo poliso XML dokumentas**

```

<sutartis>
  <kontrahentas>
    <adresai>
      <adresas>
        <salis>'LT'</salis>
        <miestas>'Vilnius'</miestas>
        <gatve>'Naugarduko.'</gatve>
        <namo_numeris>'12'</namo_numeris>
        <kontr_kodas>'43XXXXXXXX'</kontr_kodas>
      </adresas>
    </adresai>

    <kontrahento_tipas_id>6</kontrahento_tipas_id>
    <kodas>'43XXXXXXXX'</kodas>
    <vardas>'Vardas'</vardas>
    <pavarde>'Pavardė'</pavarde>
    <draudos_darb_agr>1</draudos_darb_agr>
    <salis_id>1</salis_id>
    <draudejas>1</draudejas>
  </kontrahentas>

  <polisas>
    <prasideda>to_date('2007.01.01','YYYY.MM.DD')</prasideda>
    <baigiasi>to_date('2008.01.01','YYYY.MM.DD')</baigiasi>
    <israsytas>to_date('2006.12.31','YYYY.MM.DD')</israsytas>
    <pard_vien_id>1111</pard_vien_id>
    <apdraustuju_kiekis>1</apdraustuju_kiekis>
  </polisas>

  <objektai>
    <objektas>
      <valst_nr>'XXX111'</valst_nr>
      <ident_nr>'JMZGXXXXXXXX'</ident_nr>
      <marke>'MAZDA'</marke>
      <modelis>'6'</modelis>
      <tipas>'Lengvasis'</tipas>
      <cc_mase>'nuo 1801 iki 2000 (imtinai)'</cc_mase>
    </objektas>

    <objektas>
      <rizikos_gr>'D0'</rizikos_gr>
      <regionas>'VILNIUS'</regionas>
      <zk_tipas>'A'</zk_tipas>
      <va_obj_t-zk_im_kodas>'A'</va_obj_t-zk_im_kodas>
      <im_kodas_1>'A3'</im_kodas_1>
      <pol_trukme>12</pol_trukme>
      <pard_vieta>'Kita'</pard_vieta>
    </objektas>
  </objektai>

```

```
<rizikos>
  <rizika>
    <tarifo_kodas>'rizika1</tarifo_kodas>
    <imoku_suma>160</imoku_suma>
    <draudimo_suma>345280</draudimo_suma>
  </rizika>

  <rizika>
    <tarifo_kodas>'rizika2</tarifo_kodas>
    <imoku_suma>75</imoku_suma>
    <draudimo_suma>345280</draudimo_suma>
  </rizika>
</rizikos>

<mokejimai>
  <mokejimas>
    <eil_nr>1</eil_nr>
    <eil_viso>1</eil_viso>
    <suma>161</suma>
    <data>to_date('2007.01.01','YYYY.MM.DD')</data>
  </mokejimas>
</mokejimai>
</sutartis>
```